DEVELOPMENT STANDARD

QIC-157
Revision B
16 Dec 94

ATA PACKET INTERFACE (ATAPI) FOR STREAMING TAPE

Quarter-Inch Cartridge Drive Standards, Inc.

311 East Carrillo Street
Santa Barbara, California 93101
Telephone (805) 963-3853
Fax (805) 962-1541

(See important notices on following page)

Technical Editor: Paul Boehler
Arcada Software
708 Fiero Commerce Park 10
San Luis Obispo, CA 93401
Telephone: (805) 544-1496
Internet: pboehler@arcada.com

Assistant Editor: Devon Worrell
Western Digital Corporation
8105 Irvine Center Drive
Telephone: (714) 932-7042
Internet: worrell@dt.wdc.com

Internet Forum:	idetape@dt.wdc.com	automatically reflects messages to forum
Forum Subscription:	majordomo@dt.wdc.com	send message: HELP<cr>LISTS<cr>END<cr>

Important Notices

This document is a development standard adopted by Quarter-Inch Drive Standards, Inc. (QIC). This document may be revised several times during the development cycle. It is intended solely as a guide for companies interested in developing products which can be compatible with other products developed using this document. QIC makes no representation or warranty regarding this document, and any company using this document shall do so at its sole risk, including specifically the risks that a product developed will not be compatible with any other product or that any particular performance will not be achieved. QIC shall not be liable for any exemplary, incidental, proximate or consequential damages or expenses arising from the use of this document. This development standard defines only one approach to the product. Other approaches may be available in the industry.

This development standard is an authorized and approved publication of QIC. The underlying information and materials contained herein are the exclusive property of QIC but may be referred to and utilized by the general public for any legitimate purpose, particularly in the design and development of quarter-inch cartridge tape drive subsystems. This development standard may be copied in whole or in part *provided* that no revisions, alterations or changes of any kind are made to the materials contained herein. Only QIC has the right and authority to revise or change the material contained in this development standard, and any revisions by any party other than QIC are totally unauthorized and specifically prohibited.

Compliance with this development standard may require use of one or more features covered by proprietary rights (such as features which are the subject of a patent, patent application, copyright, mask work right or trade secret right). By publication of this development standard, no position is taken by QIC with respect to the validity or infringement of any patent or other proprietary right, whether owned by a Member or Associate of QIC, or otherwise. QIC hereby expressly disclaims any liability for infringement of intellectual property rights of others by virtue of the use of this development standard. QIC has not and does not investigate any notices or allegations of infringement prompted by publication of any QIC development standard, nor does QIC undertake a duty to advise users or potential users of QIC development standards of such notices or allegations. QIC hereby expressly advises all users or potential users of this development standard to investigate and analyze any potential infringement situation, seek the advice of intellectual property counsel, and, if indicated, obtain a license under any applicable intellectual property right or take the necessary steps to avoid infringement of any intellectual property right. QIC expressly disclaims any intent to promote infringement of any intellectual property right by virtue of the evolution, adoption, or publication of any QIC development standard.

Table of Contents

I. REVISION HISTORY ix

1. ATAPI OVERVIEW 1

1.1 ATA Signal Utilization 1
1.2 ATA Command Utilization 1
1.3 ATA Compatibility 1
1.4 Packet Types 2
1.5 How SCSI is Used by ATAPI 3
1.6 Differences from the SCSI Standard 3
1.6.1 Redundant Command Functionality (Task File vs. Packet) 4
1.6.2 ATAPI Identify Device vs. INQUIRY 4
1.6.3 Initialize Drive Parameters and Set Features vs. MODE SELECT 4

2. ATAPI PROTOCOL 5

2.1 Initialization 5
2.2 PACKET Command 5
2.3 Status Register Utilization for Packet Commands 6
2.4 Byte Count Register (Cylinder Low/High) Usage for Packet Commands 6
2.5 Sector Count (ATAPI Interrupt Reason) Register Usage for Packet Commands 7
2.6 Overlapped Command Operation 7
2.7 Flow of Packet Command, PIO Data In to the Host 8
2.8 Flow of Packet Command with PIO Data Out from the Host 9
2.9 Flow of DMA Data Commands 9
2.10 Flow of Non-data Commands 10
2.11 Timing of Packet Command 11
2.12 Timing of Data and Status Transfer 12
2.13 Control Signal Timing Requirements and Relationships 12
2.14 BIOS and ATAPI Driver Compatibility 13
2.14.1 Reset Master/Slave Diagnostics Sequence 13
2.14.2 SRST Initialization Sequence 13
2.14.3 Special Handling of ATA Read and Identify Drive Commands 14
2.14.4 ATAPI aware BIOS and Driver Considerations 14
2.14.5 Default Timing 15
2.15 ATAPI Command Protocol and DSC Handling 15
2.15.1 ATAPI Media Access Commands and DSC 15
2.15.2 ATAPI Non-media Access Commands and DSC 15
2.15.3 ATAPI Read and Write Commands and DSC 15

3. ATAPI TRANSPORT MECHANISM 16

3.1 Reset Conditions 16
3.1.1 Power On or Hardware Reset 16
3.2 ATAPI Soft Reset Command and Protocol 17
3.3 ATAPI Implementation of ATA SRST 17
3.3.1 SRST Sequence 17
3.4 Physical Connection 18
3.5 Register Mapping 18
3.6 ATAPI Register Map (Packet Command) 19

4. STREAMING TAPE (TASK FILE) COMMANDS 23

4.1 ATA (Task File) Command Implementation Requirements 23
4.1.1 ATAPI Soft Reset 23
4.1.2 Check Power Mode 23
4.1.3 Execute Drive diagnostics 23
4.1.4 Idle Immediate 23
4.1.5 Nop 23

4.1.6 Packet Command *23*
4.1.7 ATAPI Identify Device *24*
4.1.8 Set Features *29*
4.1.9 Sleep *30*
4.1.10 Standby Immediate *30*

5. STREAMING TAPE DEVICE MODEL **31**

5.1 MEDIUM ATTRIBUTES 31
5.1.1 Early Warning *32*
5.1.2 Partitions *32*
5.2 QUICK FILE ACCESS 32
5.2.1 Changing Partitions *33*
5.2.2 Automatic Format Detection *33*
5.3 LOGICAL ELEMENTS WITHIN A PARTITION 34
5.3.1 Blocks *34*
5.3.2 Filemarks *34*
5.3.3 Gaps *34*
5.3.4 Blank *34*
5.4 DATA BUFFERING 35
5.5 RECORDED ELEMENT DESCRIPTORS (BLOCK IDENTIFIERS) 35
5.6 POSITIONING 36
5.6.1 Direction and Position Definitions *36*
5.7 ERROR REPORTING 37

6. STREAMING TAPE PACKET COMMANDS **38**

6.1 PACKET COMMAND IMPLEMENTATION REQUIREMENTS 38
6.1.1 Reserved *38*
6.2 ATAPI COMMAND PACKET DESCRIPTION 38
6.2.1 Operation Code *39*
6.2.2 Logical Block Address *39*
6.2.3 Transfer Length *39*
6.2.4 Parameter List Length *39*
6.2.5 Allocation Length *40*
6.3 STATUS 40
6.4 COMMAND PROCESSING CONSIDERATIONS AND EXCEPTION CONDITIONS 40
6.4.1 Parameter Rounding *40*
6.5 UNIT ATTENTION CONDITION 41
6.6 COMMANDS AND PARAMETERS 41
6.7 ATAPI MEDIA ACCESS COMMAND CONSIDERATIONS 41
6.8 ATAPI PACKET COMMANDS FOR STREAMING TAPE DEVICES 42
6.8.1 Erase Command *43*
6.8.2 INQUIRY Command *44*
6.8.3 LOAD UNLOAD Command *47*
6.8.4 LOCATE Command *49*
6.8.5 LOG SELECT Command *50*
6.8.6 LOG SENSE Command *52*
6.8.7 MODE SELECT Command *59*
6.8.8 MODE SENSE Command *61*
6.8.9 READ Command *79*
6.8.10 READ POSITION Command *81*
6.8.11 REQUEST SENSE Command *84*
6.8.12 REWIND Command *93*
6.8.13 SPACE Command *94*
6.8.14 TEST UNIT READY Command *96*
6.8.15 WRITE Command *97*
6.8.16 WRITE FILEMARK Command *99*
6.9 GLOSSARY FOR ATAPI DEVICE 100

6.10 VENDOR IDENTIFICATION LIST 101

FIGURE 3-1 ATAPI STATUS REGISTER (ATA STATUS REGISTER) 20
FIGURE 3-2 ATAPI ERROR REGISTER (ATA ERROR REGISTER) 20
FIGURE 3-3 ATAPI FEATURE REGISTER (ATA FEATURE REGISTER) 20
FIGURE 3-4 ATAPI BYTE COUNT REGISTER (ATA CYLINDER HIGH/LOW REGISTER) 21
FIGURE 3-5 ATAPI INTERRUPT REASON REGISTER (ATA SECTOR COUNT REGISTER) 21
FIGURE 3-6 ATAPI DRIVE SELECT REGISTER (ATA DRIVE / HEAD SELECT REGISTER) 22
FIGURE 3-7 ATAPI DEVICE CONTROL REGISTER (ATA DRIVE CONTROL REGISTER) 22

TABLE -1-1 COMMAND AND STATUS USAGE FOR ATAPI DEVICES (TAPE) 2
TABLE 2-1 BYTE COUNT REGISTER USAGE 7
TABLE 3-1 PREFERRED DEVICE CONNECTION 18
TABLE 3-2 I/O PORT FUNCTIONS/SELECTION ADDRESSES (COMPATIBILITY MODEL) 19
TABLE 3-3 WORD MAPPING OF TASK FILE REGISTERS 19
TABLE 4-1 IDENTIFY DEVICE, FIELDS SUPPORTED BY STREAMING TAPE 24
TABLE 4-2 IDENTIFY DEVICE DATA - GENERAL CONFIGURATION (WORD 0) 25
TABLE 4-3 IDENTIFY DEVICE DATA - CAPABILITIES WORD (49) 26
TABLE 4-4 CONTENTS OF THE FEATURE REGISTER FOR SET FEATURES COMMAND 29
TABLE 4-5 SET FEATURE REGISTER DEFINITIONS 29
TABLE 4-6 FEATURE NUMBER DESCRIPTION FOR SET FEATURE COMMAND 30
TABLE 6-1 TYPICAL COMMAND PACKET FOR MOST COMMANDS 38
TABLE 6-2 TYPICAL COMMAND PACKET FOR READ OR WRITE OPERATIONS 39
TABLE 6-3 OPERATION CODE 39
TABLE 6-4 PACKET COMMANDS SUPPORTED BY STREAMING TAPE DEVICES 42
TABLE 6-6 ERASE COMMAND 43
TABLE 6-7 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR ERASE ERRORS 43
TABLE 6-8 INQUIRY COMMAND 44
TABLE 6-9 INQUIRY DATA FORMAT 45
TABLE 6-10 PERIPHERAL DEVICE TYPES 45
TABLE 6-11 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR INQUIRY ERRORS 46
TABLE 6-12 LOAD UNLOAD COMMAND 47
TABLE 6-13 LOAD, EOT, AND RETENSION BIT COMBINATIONS 48
TABLE 6-14 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR LOAD UNLOAD ERRORS 48
TABLE 6-15 LOCATE COMMAND 49
TABLE 6-16 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR LOCATE ERRORS 49
TABLE 6-17 LOG SELECT COMMAND 50
TABLE 6-18 PAGE CONTROL FIELD 50
TABLE 6-19 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR LOG SELECT ERRORS 51
TABLE 6-20 LOG SENSE COMMAND 52
TABLE 6-21 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR LOG SENSE ERRORS 53
TABLE 6-22 LOG PAGE CODES 54
TABLE 6-23 LOG PAGE FORMAT 54
TABLE 6-24 LOG PARAMETER 55
TABLE 6-25 SUPPORTED LOG PAGE 55
TABLE 6-26 PARAMETER CODES FOR ERROR COUNTER PAGES 56
TABLE 6-27 PARAMETER CODES FOR ERROR COUNTER PAGE (READ) - PAGE CODE 03H 56
TABLE 6-28 PARAMETER CODES FOR ERROR COUNTER PAGE (WRITE) - PAGE CODE 02H 57
TABLE 6-29 TAPE CAPACITY PAGE CODE 31H 58
TABLE 6-30 MODE SELECT COMMAND 59
TABLE 6-31 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR MODE SELECT ERRORS 60
TABLE 6-32 MODE SENSE COMMAND 61
TABLE 6-33 PAGE CONTROL FIELD 61
TABLE 6-34 PAGE CODE USAGE FOR ALL DEVICES 61
TABLE 6-35 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR MODE SENSE ERRORS 62
TABLE 6-36 MODE PAGE CODES 63

TABLE 6-37 MODE PARAMETER LIST....63
TABLE 6-38 MODE PARAMETER HEADER....63
TABLE 6-39 MODE PARAMETER BLOCK DESCRIPTOR....64
TABLE 6-40 MODE PAGE FORMAT....64
TABLE 6-41 ATAPI STREAMING TAPE DEVICE MEDIUM TYPE CODES....67
TABLE 6-42 KEY FOR MEDIUM TYPE CODES....68
TABLE 6-43 KEY FOR 5.25" CARTRIDGES:....69
TABLE 6-44 KEY FOR 3.5" CARTRIDGES:....69
TABLE 6-45 DEVICE-SPECIFIC PARAMETER....69
TABLE 6-46 SPEED FIELD ASSIGNMENTS....69
TABLE 6-47 ATAPI STREAMING TAPE DEVICE DENSITY CODES....71
TABLE 6-48 KEY FOR DENSITY CODES....72
TABLE 6-49 DATA COMPRESSION PAGE....72
TABLE 6-50 POSSIBLE DATA COMPRESSION BOUNDARIES....73
TABLE 6-51 BOUNDARIES THAT GENERATE A "CHECK CONDITION" WHEN RED FIELD IS ZERO....73
TABLE 6-52 BOUNDARIES THAT GENERATE A "CHECK CONDITION" WHEN RED FIELD IS ONE....74
TABLE 6-53 BOUNDARIES THAT GENERATE A "CHECK CONDITION" WHEN RED FIELD IS ZERO....74
TABLE 6-54 COMPRESSION ALGORITHM IDENTIFIER....76
TABLE 6-55 MEDIUM PARTITION PAGE....76
TABLE 6-56 MEDIUM FORMAT RECOGNITION FIELD....77
TABLE 6-57 CAPABILITIES AND MECHANICAL STATUS PAGE....77
TABLE 6-58 READ COMMAND....79
TABLE 6-59 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR READ ERRORS....80
TABLE 6-60 READ POSITION PHYSICAL BLOCK ADDRESSING HAS [i]BEEN REMOVED[ii] THE MEDIUM PARTITION MODE PAGE HAS BEEN SIMPLIFIED. COMMAND....81
TABLE 6-61 READ POSITION DATA FORMAT....82
TABLE 6-62 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR READ POSITION ERRORS....83
TABLE 6-63 REQUEST SENSE COMMAND....84
TABLE 6-64 REQUEST SENSE STANDARD DATA....85
TABLE 6-65 FIELD POINTER BYTES....87
TABLE 6-66 FIELD POINTER BYTES....87
TABLE 6-67 SENSE KEY DESCRIPTIONS....89
TABLE 6-68 ASC AND ASCQ ASSIGNMENTS....90
TABLE 6-69 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR REQUEST SENSE ERRORS....92
TABLE 6-70 REWIND COMMAND....93
TABLE 6-71 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR REWIND ERRORS....93
TABLE 6-72 SPACE COMMAND....94
TABLE 6-73 SPACE CODE....94
TABLE 6-74 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR SPACE ERRORS....95
TABLE 6-75 TEST UNIT READY COMMAND....96
TABLE 6-76 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR TEST UNIT READY ERRORS....96
TABLE 6-77 WRITE COMMAND....97
TABLE 6-78 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR WRITE ERRORS....98
TABLE 6-79 WRITE FILEMARK COMMAND....99
TABLE 6-80 RECOMMENDED SENSE KEY, ASC AND ASCQ FOR WRITE FILEMARK ERRORS....99

I. Revision History

Changes for Revision B:

Many clarifications and simplifications were made throughout the specification. Usage of drive seek complete (DSC) line and byte count register have been clarified. All transfer lengths are a multiple of 4 bytes. The incomplete floppy backward compatible model has been removed. Physical block addressing has been removed. The read position command has been simplified. Various mode pages have been removed and only current values are reportable. Various log pages have been removed and logging capabilities have been eliminated such as save parameters and various page control options. It should be noted that items not documented in this specification can be implemented, but are not considered a required element of an ATAPI drive.

1. ATAPI Overview

The purpose of the ATAPI is to provide a more extensible and general purpose interface than the ATA Task file.

Although the attachment of a Streaming Tape on the ATAPI Interface will utilize the ATA Host Hardware and Task File, the logical interface will differ slightly and will need to support additional capabilities. The Mass Storage devices connected to the ATA make use of eight registers (Task File) that contain the command and all parameters needed for operation. However, eight registers is not enough to pass all the needed information for commanding other peripheral types. To remedy this, the ATAPI Device will receive its commands through the use of a Packet mode, in addition to the normal ATA protocol. The Packet Command will complement the existing ATA commands. The ATAPI Device shall support all of the ATA specified protocol, including the Reset Master/Slave Diagnostic Sequence, Diagnostic Command, and Command Abort for unsupported Commands. The ATAPI Device shall also support both the Master and Slave modes of operation.

1.1. ATA Signal Utilization

ATAPI Devices shall utilize the same signals and timing from the ATA Standard and Extensions.

1.2. ATA Command Utilization

The ATA Task File concept does not contain enough bytes to support some of the Streaming Tape command structures, so a new command called “Packet Command” has been added to allow a Packet to be sent to the Device. The Packet shall be transferred by writing multiple times to the Data Register. No random access to the register file in the Peripheral can be done. This technique reduces the number of register addresses needed, but not the actual space needed. Although all the commands for the Device could be sent via this packet mode, it is believed that some of the existing ATA commands and the full ATA command protocol must be provided for the existing drivers to operate correctly. The Device will therefore support some existing ATA commands in addition to the new “Packet command”, so that there will be minimal changes to the existing drivers. This minimal set of ATA commands is different than the minimum as defined in the ATA standard, but should be sufficient for normal operation.

1.3. ATA Compatibility

There are several backward compatibility issues with the existing ATA commands, and therefore the ATAPI Device shall respond to the existing ATA Reset Master/Slave Diagnostic Sequence, but not the Identify Drive or Read commands. This will allow the BIOS and older drivers to ignore the ATAPI Device and not confuse ATAPI data with normal ATA Drive format data. All unsupported ATA commands shall be Aborted, and not executed. As with aborted commands in ATA, an interrupt shall be generated to signal the completion with an “Aborted” error status.

1.4. Packet Types

To allow for generic packet transfer and the connection of SCSI like peripherals, there shall exist a minimum set of information that is exchanged. This information shall generically support the following:

- Command Packet (Always padded to number of bytes identified in byte 0 of the Identify Device data. 00 = 12 bytes, 01 = 16 bytes)
- Command Parameter Data (e.g. Write Data etc.)
- Command Response Data (e.g. Read Data etc.)
- Status. The Status will not take the form of a packet of information. The status will be presented using the ATAPI Status Register (redefinition's of the ATA Status Register).

Table -1-1 Command and Status Usage for ATAPI Devices (Tape)

Command	Used	Code	Error Register						Status Register				
			SK1	SK0	MCR	ABRT	EOM	ILI	DRDY		DSC		ERR
Acknowledge media change	N	DB				V							V
Boot - post-boot	N	DC				V							V
Boot - pre-boot	N	DD				V							V
Check power mode	M	E5				V			V		V		V
Door lock	N	DE				V							V
Door unlock	N	DF				V							V
Execute Drive diags	M	90	Special Device Diagnostic Errors										V
Format track	N*	50				V			V		V		V
Identify Drive	N	EC				V							V
Idle	N	E3				V			V		V		V
Idle immediate	M	E1				V			V		V		V
Initialize Drive parms	N*	91				V							V
NOP	M	00				V							V
ATAPI Pkt. Command	M	A0	Contains Packet Command Status						V		V		V
ATAPI Identify Device	M	A1				V			V		V		V
ATAPI Soft Reset	M	08											
Read buffer	N	E4				V							V
Read DMA (w/retry)	N	C8				V							V
Read DMA (wo/retry)	N	C9				V							V
Read long (w/retry)	N*	22				V							V
Read long (wo/retry)	N*	23				V							V
Read multiple	N	C4				V							V
Read sector(s) (w/retry)	N*	20				V							V
Read sector(s) (wo/retry)	N*	21				V							V
Read verify sector(s) (w/retry)	N*	40				V							V
Read verify sector(s) (wo/retry)	N*	41				V							V
Recalibrate	N*	1x				V			V		V		V
Seek	N	7x				V							V
Set features	M	EF				V			V		V		V
Set multiple mode	N	C6				V			V		V		V
Sleep	M	E6				V			V		V		V
Standby	N	E2				V			V		V		V
Standby immediate	M	E0				V			V		V		V
Write buffer	N	E8				V							V
Write DMA (w/retry)	N	CA				V							V

Write DMA (wo/retry)	N	CB				V							V
Write long (w/retry)	N*	32				V							V
Write long (wo/retry)	N*	33				V							V
Write multiple	N	C5				V							V
Write same	N	E9				V							V
Write sector(s) (w/retry)	N*	30				V							V
Write sector(s) (wo/retry)	N*	31				V							V
Write verify	N	3C				V							V
Invalid command code						V			V		V		V

V = valid on this command
M = Mandatory and shall be supported by ATAPI Devices, as specified by the ATA Standard
O = Optional for use by an ATAPI Device
N = Not supported by ATAPI Devices (shall be Aborted by the ATAPI Device)
* The ATA Standard specifies command as Mandatory

1.5. How SCSI is Used by ATAPI

Although the ATAPI Device will utilize many of the actual packet definitions from the SCSI standard, it will NOT use most other features of the normal SCSI Protocol. Thus there are no Phases, no Messages, no sharable bus, (only one Host Computer) and no SCSI Hardware. For those who are familiar with the current SCSI-3 effort, this specification will not conform with that Packetized Standard, due mostly to limitations of ATAPI.

1.6. Differences from the SCSI Standard

Some of the major differences from the SCSI Standard:

- Status shall use the ATAPI description, rather than a Data Byte passed at the end of the command.
- ATAPI Device is slave during operation rather than the master view of a SCSI Peripheral.
- No messages are supported.
- No disconnect/reconnect or any of the SCSI Pointers.
- No linking or queuing of commands.
- All Streaming Tape Command Packets (CP) are 12 bytes in length, rather than the 6, 8, 10 or 12-byte packets of the SCSI Standard; however, 16-byte ATAPI Command Packets are defined for future Devices. The size of the Command Packet required by a Device is defined in word 0 of the ATAPI Identify Device command, allowing Host Device Drivers to determine the size of the Command Packets before issuing an ATAPI Command Packet.
- No allegiance conditions are used.

This specification will make use of many of the Standard SCSI Command Descriptor Block definitions and Commands, but some of the commands that would normally be supported by a SCSI Device will not be supported for various reasons. These commands are:

- RESERVE and RELEASE; as there is only one Host allowed, this is not needed.
- CHANGE DEFINITIONS; as there is no SCSI, this command is nonsensical.
- COPY, COPY/VERIFY; no shared bus so this command cannot be implemented.

1.6.1. Redundant Command Functionality (Task File vs. Packet)

The SCSI Standard has provided some commands that the ATA Standard also provides. It is the intent of this specification to allow all the functionality to exist, by utilizing only Command Packets. This will allow existing SCSI like Drivers to continue to issue packets for all operation, and has some lower level Driver converting them to the ATAPI protocol.

1.6.2. ATAPI Identify Device vs. INQUIRY

The ATAPI Identify Device command has information that the low level Drivers use to perform ATA interface hardware configuration. Information in the Identify Device shall continue to look exactly as the ATA Identify Drive does for compatibility reasons. As the information in the INQUIRY Command cannot be returned by the ATAPI Identify Device Command, the INQUIRY Command shall be supported for use by higher level drivers.

1.6.3. Initialize Drive Parameters and Set Features vs. MODE SELECT

The Initialize Drive Parameters command does not contain a method to provide non ATA drive configuration information, and shall not be used. As such, the MODE SELECT and MODE SENSE from the SCSI standard shall be supported. The combination of MODE SELECT and Set Features commands contains all the necessary functionality and is most compatible with the existing BIOS's and OS Drivers.

2. ATAPI Protocol

The ATAPI Device is commanded by two methods, the original ATA Command protocol utilizing the Task File and the new Packet Command method. For both methods, the Devices using this interface shall be programmed by the Host to perform commands and return status to the Host at command completion. When more than one Device is daisy chained on the interface, commands are written in parallel to all peripherals, and for ATA commands, except the Execute Diagnostics command, only the selected Device (DRV bit in the Drive/Head ATA Register) executes the command. On an Execute Diagnostics command addressed to Device 0, both Devices shall execute the command, and Device 1 shall send its status to Device 0 via PDIAG-.

The “Protocol” for ATAPI centers around the usage of a new ATA Command called “Packet Command.” All the normal ATA rules and protocol are used to issue the Packet Command, but once the command has been issued a new set of rules applies:

1. The interpretation of the DRQ bit in the Status Register shall be used along with the Interrupt Reason Registers to determine the actual Interrupt Type.
2. The actual command for the Device to execute is sent as a packet via the data register, and not the Task File.
3. Command arguments are supplied by the Command Packet as well as from the Task File.
4. A Byte Count is used to determine the amount of data the Host shall transfer at each DRQ Interrupt.
5. The ATAPI Features Register is used to indicate when DMA will be used rather than by using different opcodes.
6. The final status is presented to the Host as a new interrupt after the last data has been transferred, rather than along with the last block of data.

These new rules (protocol) only apply from after the issuance of the Packet Command, until the Completion Status has been read by the Host. After the Completion Status has been read, the Task File Register definitions and Protocol revert to the standard ATA definition.

2.1. Initialization

The ATAPI Device shall respond just as defined in the ATA Standard. The DASP and PDIAG signals shall be utilized following any reset condition, except the ATAPI Soft Reset command.

2.2. PACKET Command

The Packet Command is issued exactly as normal ATA commands, by initializing the Task File Registers, setting the Device Selection Bit and writing the Command byte into the Command Register. With normal ATA commands a DRQ (Optional Interrupt) would be generated to indicate that the data for the command could be transferred to/from the Device. With the Packet Command, the first DRQ indicates that the Command Packet Data shall be written to the Device. Once the Command Packet has been sent, the command proceeds as a normal ATA command would. The Command Packet bytes shall always be transferred via PIO and never using DMA.

ATA Packet Commands can be issued regardless of the state of the DRDY Status Bit.

If, while polling BSY, the Device remains in a state where it cannot accept a command for more than 5 seconds, the Host shall time out and reset the Device.

Data transfers may be accomplished in more ways than are described by this specification, but the sequences described in this specification shall be used to remain compatible with current and future ATAPI Devices.

2.3. Status Register Utilization for Packet Commands

See 3.6 ATAPI Register Map (Packet Command) on page 19 for a description of the register definition.

2.4. Byte Count Register (Cylinder Low/High) Usage for Packet Commands

This register is used to control the number of bytes the Host shall transfer at each DRQ. It is only used for the command parameter data being transferred via PIO and never for DMA or Command Packet bytes.

Since the length of data that is actually transferred to and from an ATAPI Device using PIO is controlled by the Host, and since the ATAPI Device needs to be able to control the number of bytes transferred, an additional capability was needed. By using the Byte Count Register, a capability to transfer a variable number of bytes has been created. In ATAPI the Device indicates to the Host the number of bytes that shall be transferred on each DRQ Interrupt. Before transferring data, the Host shall read the 16-bit Byte Count Register, and comply with the length requested. Both the ATAPI Device and the Host will have their own byte counts and will transfer until those counts go to zero. For some commands, such as MODE SENSE, the Host does not know the amount of data that will be transferred, and shall rely on the Byte Count supplied by the Device to transfer the correct amount of data.

When using DMA to transfer data, the Byte Count Register is not used and will contain unspecified data.

A further capability of the Byte Count Register is for the Host to signal to the ATAPI Device the maximum amount of data it can take in a single PIO DRQ packet or the preferred PIO DRQ packet size. For all commands that require data be transferred, the Host shall set the Byte Count Register to the desired length before issuing the Packet Command. This length shall be used by the ATAPI Device as the maximum size for each PIO data packet. The Device can choose to transfer PIO DRQ packets smaller than those set by the Host in the Byte Count Register. The smallest maximum byte count that the host can specify is 512, and devices that transfer 512 bytes or less per DRQ may ignore this field.

For all commands that can transfer all the data in one DRQ Interrupt, the Byte Count shall contain the total data length. When a Read command is being processed, the ATAPI Device may wish to send all the data that is available in its buffers on just one DRQ Interrupt, with the limitation that only 65535 bytes may be transferred at one time.

Table 2-1 Byte Count Register Usage

Operation	Usage (PIO)	Usage (DMA)
Send Command Packet	Is used as a parameter to the Packet Command and is not used to control the Packet transfer.	Command Packet is always sent via Programmed I/O and not DMA.
Parameters to the Packet Command (Task File Contents)	As a parameter to any Packet Command that will transfer ata, the Byte Count is used by the Host to communicate the maximum / preferred amount of data to be transferred on each DRQ.	The Device can ignore the byte count, as the actual transfers are controlled via the ATAPI Device and not the Host.
Parameter Data from the Device to the Host (e.g. data from a READ, or INQUIRY command)	At each DRQ the count contains the number of bytes that the Host shall transfer from the Device.	The ATAPI Device can transfer data whenever it wishes, and as such the Byte Count shall not be used.
Parameter Data from the Host to the Device (e.g. data for a Write, or Mode Select command)	At each DRQ the count contains the number of bytes that the Host shall transfer to the Device.	The ATAPI Device can transfer data whenever it wishes, and as such the Byte Count shall not be used.

If the Device requests more data transferred than required by the command protocol, the Host shall pad when sending data to the Device, and dump extra data into a bit bucket when reading data from the Device. Only the amount of data specified or implied in an ATAPI packet shall be transferred; transfer beyond this may cause data corruption.

On odd byte transfers, the only permissible time for an actual Odd Byte Count value will be on the Last DRQ. Intermediate DRQs shall contain even byte counts.

2.5. Sector Count (ATAPI Interrupt Reason) Register Usage for Packet Commands

The Interrupt Reason Register contains an expanded definition of the ATA DRQ Status. When the DRQ is presented in the Status Register for an ATAPI Packet Command, then the contents of this register indicate if Packet Command or User Data shall be transferred and the direction of transfer.

2.6. Overlapped Command Operation

ATAPI media access commands have immediate status with a delayed process. These commands return Completion Status immediately with the actual execution of the command continuing. For these commands, the DSC bit is cleared until actual completion has finished. If a new command is received before DSC is set, the drive will respond in one of the following ways:

- The new command is queued and BSY is left set; or
- The new command is executed with no impact on the previous command.

1. ATA commands operate differently from packet commands. When a new ATA command is written to the Command Register while BSY bit is set, the executing command stops execution and reports a “Check Condition,” “Aborted,” and ABORTED COMMAND Sense Key.
2. New ATAPI media access commands received while the DSC bit is cleared shall cause the BSY signal to be held until the previous command completes. New ATAPI non-media access commands will complete normally without impacting any previous command. If a check condition exists due to a previous command, that condition will be reported as a deferred error

upon receipt of any new ATAPI command. (See 2.15 ATAPI Command Protocol and DSC Handling , page 5.)

3. If a command is executing when the Device is issued a SRST, the DSC bit shall be cleared with the rest of the Status Register. The Host must issue ATAPI Identify or any packet command to restore DSC operation.

2.7. Flow of Packet Command, PIO Data In to the Host

This class includes commands such as INQUIRY, READ etc. Execution includes the transfer of some unknown number of data bytes from the Device to the Host. Note that ATAPI Identify is not a packet command and uses ATA protocol (not ATAPI).

1. The Host Polls for BSY=0, DRQ=0 then initializes the task file by writing the required parameters to the Features, Byte Count, and Drive/Head registers.
2. The Host writes the Packet Command code (A0h) to the Command Register.
3. The Device sets BSY, before the next system read of the status register, and prepares for Command Packet transfer.
4. When the Device is ready to accept the Command Packet, the Device sets CoD and clears IO, BSY prior to asserting DRQ. Some Devices will assert INTRQ following the assertion of DRQ. See section 4.1.7.1 General Configuration Word (0), on page 25 for command packet DRQ types and other related timing information. DRQ may be set before or after BSY has been de-asserted; however, DRQ will not be visible to the Host until BSY=0.
5. After detecting DRQ, the Host writes the 12 bytes (6 words) of Command to the Data Register.
6. The Device(1) clears DRQ (when the 12^{th} byte is written), (2) sets BSY, (3) reads Features and Byte Count requested by the Host system, (4) prepares for data transfer.
7. When data is available, the Device: (1) places the byte count of the data available into the Cylinder High and Low Registers, (2) sets IO and clears CoD, (3) sets DRQ and clears BSY, (4) sets INTRQ.
8. After detecting INTRQ, the Host reads the DRQ bit in the Status Register to determine how it shall proceed with the command. If DRQ= 0 then the Device has terminated the command. If DRQ=1 then the Host shall read the data (number of bytes specified in the Cylinder High/Low Registers) via the Data Register. In response to the Status Register being read, the Device negates INTRQ for both cases.
9. The Device clears DRQ. If transfer of more data is required, the Device also sets BSY and the above sequence is repeated from step 7.
10. When the Device is ready to present the status, the Device places the completion status into the Status Register, sets CoD, IO, DRDY and clears BSY, DRQ, prior to asserting INTRQ.
11. After detecting INTRQ & DRQ=0, the Host reads the Status Register and, if necessary, the Error Register for the command Completion Status.

The DRQ signal is used by the Device to indicate when it is ready to transfer data, and is cleared after (during) the last byte of data to be transferred. This applies for Command Packet as well as normal read/write data.

2.8. Flow of Packet Command with PIO Data Out from the Host

This class includes commands such as Mode Select, Write etc. Execution includes the transfer of some known number of data bytes from the Host to the Device.

1. The Host Polls for BSY=0, DRQ=0 then initializes the task file by writing the required parameters to the Features, Byte Count, and Drive/Head registers.
2. The Host writes the Packet Command code (A0h) to the Command Register.
3. The Device sets BSY, before the next system read of the status register, and prepares for Command Packet transfer.
4. When the Device is ready to accept the Command Packet, the Device sets CoD and clears IO, BSY prior to asserting DRQ. Some Devices will assert INTRQ following the assertion of DRQ. See section 4.1.7.1 General Configuration Word (0), on page 25 for command packet DRQ types and other related timing information.
5. After detecting DRQ, the Host writes the 12 bytes (6 words) of Command to the Data Register.
6. The Device(1) clears DRQ (when the 12^{th} byte is written), (2) sets BSY, (3) reads Features and Byte Count requested by the Host system, (4) prepares for data transfer.
7. When ready to transfer data, the Device:(1) sets the byte count (Cylinder High and Low Registers) to the amount of data that the Device wishes to be sent , (2) clears IO and CoD, (3) sets DRQ and clears BSY, (4) sets INTRQ. The Byte Count would normally be set to the number of bytes requested by the contents of the register at the receipt of the command, but may be any amount that the Device can accommodate in its buffers at this time.
8. After detecting INTRQ, the Host reads the DRQ bit in the Status Register to determine how it shall proceed with the command. If DRQ= 0, then the Device has terminated the command. If DRQ=1, then the Host shall write the data (number of bytes specified in the Cylinder High/Low Registers) via the Data Register. In response to the Status Register being read, the Device negates INTRQ for both cases.
9. The Device clears DRQ and sets BSY. If transfer of more data is required, the above sequence is repeated from 7.
10. When the Device is ready to present the status, the Device places the completion status into the Status Register, sets CoD, IO, DRDY and clears BSY, DRQ, prior to asserting INTRQ.
11. After detecting INTRQ & DRQ=0, the Host reads the Status Register and if necessary, the Error Register for the command Completion Status.

The DRQ signal is used by the Device to indicate when it is ready to transfer data, and is cleared after (during) the last byte of data to be transferred. This applies for Command Packet as well as normal read/write data.

2.9. Flow of DMA Data Commands

This class includes commands such as Read, Write etc. Execution includes the transfer of some unknown number of data bytes.

1. The Host Polls for BSY=0, DRQ=0 then initializes the task file by writing the required parameters to the Features, Byte Count, and Drive/Head registers. The Host must also initialize the DMA engine that will service the Device requests.

2. The Host writes the Packet Command code (A0h) to the Command Register.
3. The Device sets BSY and prepares for Command Packet transfer.
4. When the Device is ready to accept the Command Packet, the Device sets CoD and clears IO, BSY prior to asserting DRQ. Some Devices will assert INTRQ following the assertion of DRQ. See section 4.1.7.1 General Configuration Word (0), on page 25 for command packet DRQ types and other related timing information.
5. After detecting DRQ, the Host writes the 12 bytes (6 words) of Command to the Data Register.
6. The Device (1) clears DRQ (when the 12th byte is written), (2) sets BSY, (3) reads Features and Byte Count requested by the Host system, (4) prepares for data transfer.
7. When ready to transfer data, the Device transfers via DMARQ/DMACK any amount that the Device can accommodate or has in its buffers at this time. This continues until all the data has been transferred.
8. When the Device is ready to present the status, the Device places the completion status into the Status Register, and sets IO, CoD, DRDY and clears BSY, DRQ, prior to asserting INTRQ.

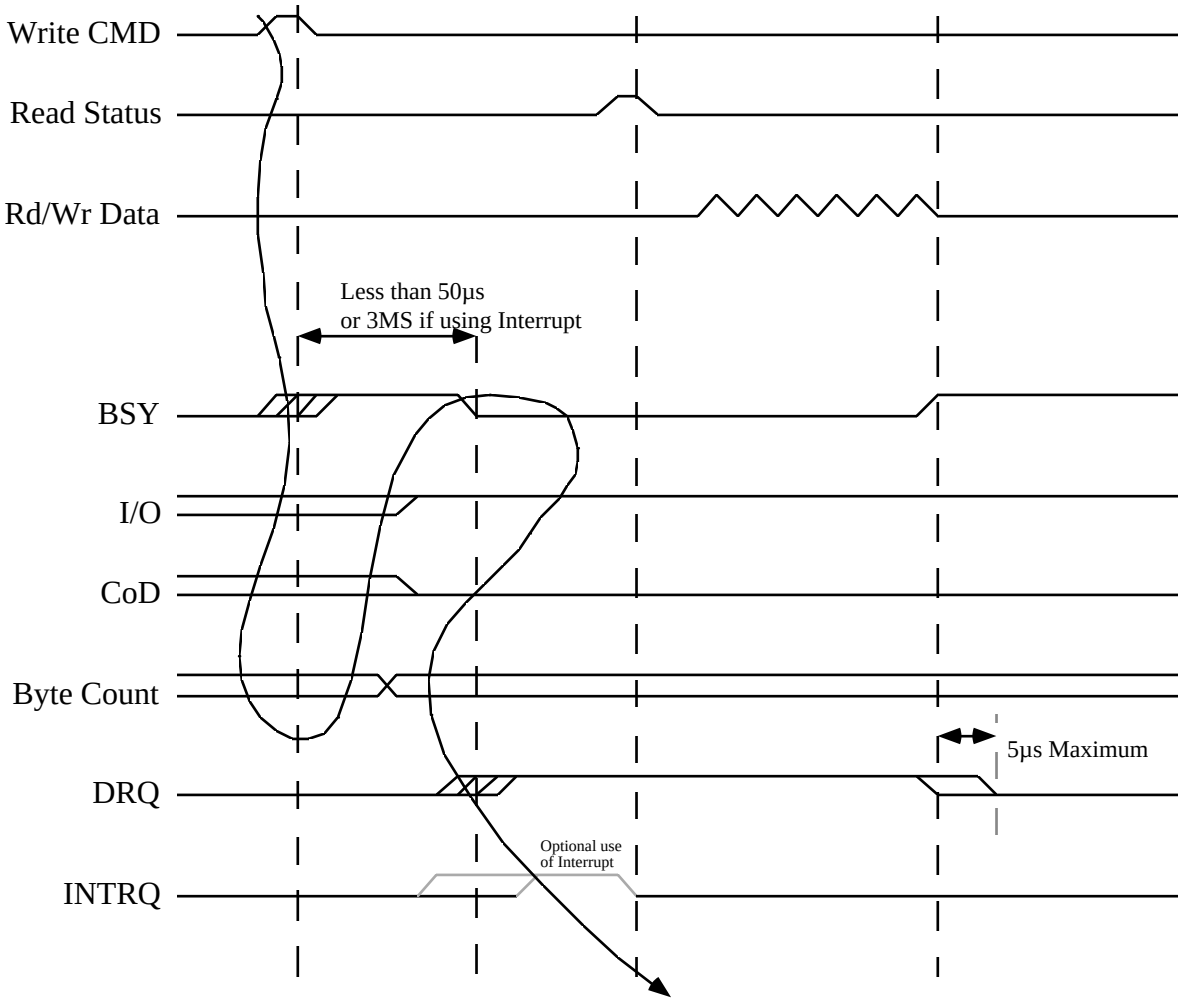
After detecting INTRQ, the Host reads the Status Register for the command Completion Status.

2.10. Flow of Non-data Commands

This class includes commands such as Rewind, etc. Execution of these commands involves no data transfer.

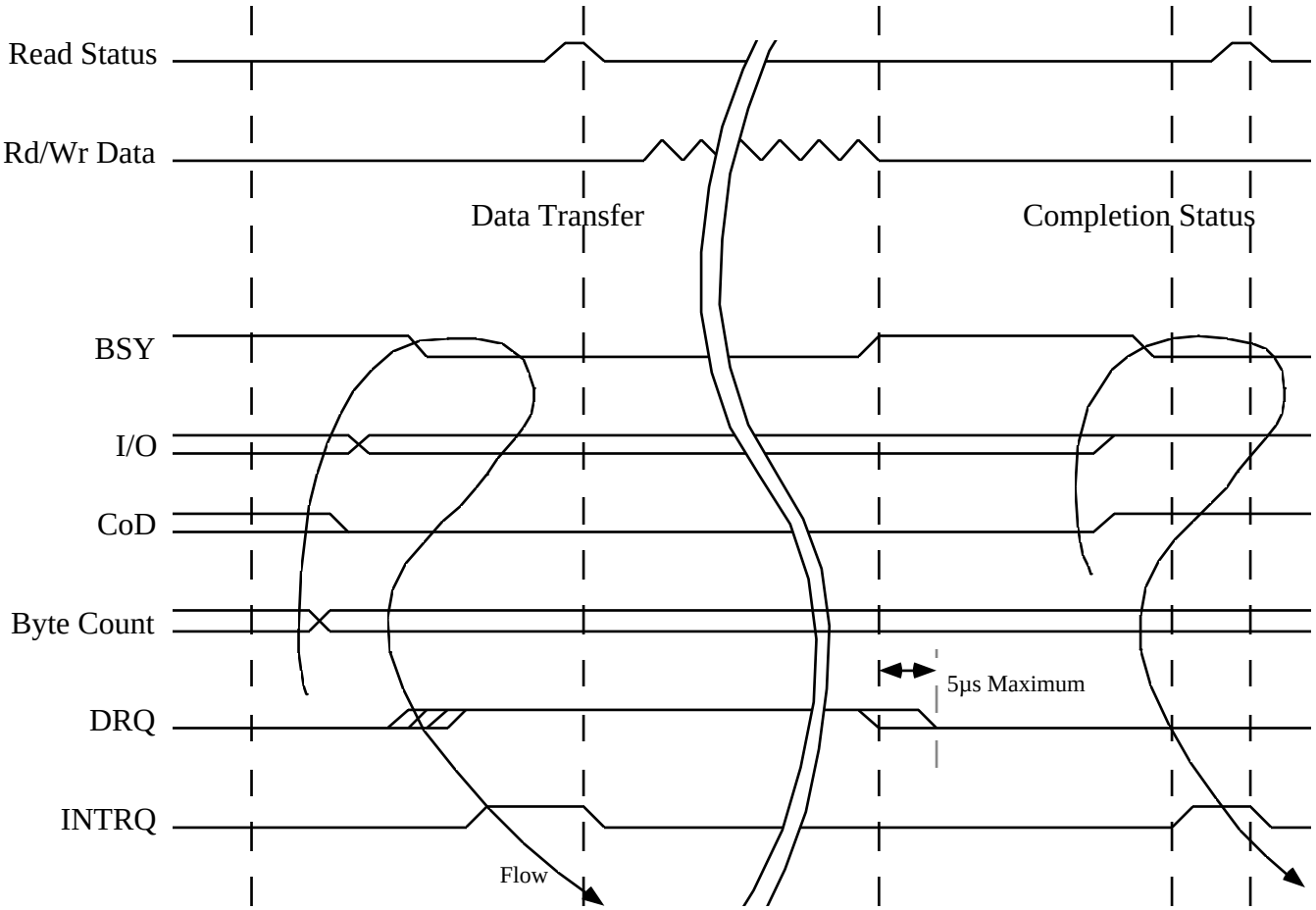
1. The Host Polls for BSY=0, DRQ=0 then initializes the task file by writing the required parameters to the Features, Byte Count, and Drive/Head registers.
2. The Host writes the Packet Command code (A0h) to the Command Register.
3. The Device sets BSY and prepares for Command Packet transfer.
4. When the Device is ready to accept the Command Packet, the Device sets CoD and clears IO, BSY prior to asserting DRQ. Some Devices will assert INTRQ following the assertion of DRQ. See section 4.1.7.1 General Configuration Word (0), on page 25 for command packet DRQ types and other related timing information.
5. After detecting DRQ, the Host writes the 12 bytes (6 words) of Command to the Data Register.
6. The Device sets BSY and executes the command.
7. When the Device is ready to present the status, the Device places the completion status into the Status Register, and sets IO, CoD, DRDY and clears BSY, DRQ, prior to asserting INTRQ.
8. After detecting INTRQ, the Host reads the Status Resister for the command Completion Status.

2.11. Timing of Packet Command



[**Editor's Note:** Time from Write CMD to BSY=0 should be:
"Less than 50 us, 3ms
or 10ms if using Interrupt"
I don't currently have the ability to change this drawing]

2.12. Timing of Data and Status Transfer



2.13. Control Signal Timing Requirements and Relationships

The order that the signals change shall adhere to the following conditions:

1. Upon receiving the A0h ATAPI Packet Command the Device shall have BSY asserted until the next Host access of the Status Register where the Device can guarantee that CoD=1 and IO=0.
2. The Device shall not assert DRQ until CoD and IO are valid for the command or data packet to be transferred and the Device is ready to perform that transfer.
3. DRQ may be set before or after BSY has been deasserted.
4. The Device shall clear BSY and set DRQ within the time-out specified by the CMD DRQ Type. See section 4.1.7.1 General Configuration Word (0), on page 25 for additional information.
5. Devices reporting CMD DRQ Type “Accelerated” shall de-assert DRQ within 5us of the last word transferred for a command or data packet.
6. Devices reporting a CMD DRQ Type other than “Accelerated” shall de-assert DRQ, before asserting INTRQ, following the last word transferred for a command or data packet.

Implementor's Note: *Early ATAPI Devices reporting CMD DRQ Types other than “Accelerated” may not be able to de-assert DRQ before the next INTRQ. Host systems should therefore wait until*

the Device asserts INTRQ before testing DRQ following the transfer of the last data word in a command or data packet.

See section 4.1.7.1 General Configuration Word (0), on page 25 for additional DRQ and other timing related information.

2.14. BIOS and ATAPI Driver Compatibility

This section discusses the IDE features and functions that shall be provided by the ATA Drive to allow the BIOS and Driver to be content.

2.14.1. Reset Master/Slave Diagnostics Sequence

A Reset Master/Slave Diagnostics Sequence with a “Good” Status shall be provided or the BIOS will not continue. When the Streaming Tape is the slave Device, and it does not respond after the Reset or Diagnostic Commands, the Master Device will return an Error Condition to the Host and all will die.

2.14.2. SRST Initialization Sequence

The SRST bit in the ATAPI Device Control Register (See Figure 3-7 ATAPI Device Control Register (ATA Drive Control Register) on page 22) shall NOT be used by the ATAPI Driver. Instead the ATAPI Device Driver shall reset the ATAPI Device utilizing the ATAPI Soft Reset command (see 3.2 ATAPI Soft Reset Command and Protocol on page 17). Resetting the ATAPI Device using the ATA SRST would also reset any ATA hard Drive attached, and if there are separate Drivers for an IDE and an ATAPI Device, each Driver would be resetting the others peripheral without the other Driver being aware of the reset.

After Receipt of an ATAPI Packet Command there are several differences from the ATA Specification:

A value other than 00h in the status register prior to the receipt of the first ATAPI Command Packet from the Host may cause ATAPI Devices to be incorrectly identified by pre-ATAPI Host BIOS as an ATA-compatible disk Drive.

Initializing the task file upon receipt of an SRST should work since only immediate commands with a delayed process will be executing when an ATA disk Driver issues an SRST. To prevent interruption of the delayed process that have not finished executing, the function of the DSC bit (i.e. command complete) shall be restored (along with DRDY being set) by issuing an ATAPI Identify or Packet command. On a warm boot the BIOS and/or Drivers will see a status of 00h.

The signature placed in the task file following an SRST shall remain until the ATAPI Device receives its first ATAPI command, i.e., the ATAPI Device will look NOT READY (DRDY=0). This will not affect the ATAPI Device Drivers ability to send ATAPI commands to the ATAPI Device since it is not required to wait for DRDY=1. However, it will prevent ATA-compatible Drivers, such as those performing power management, from sending commands to an ATAPI Device until the ATAPI Device has received its first ATAPI command: ATAPI Packet Command, ATAPI Identify Device.

ATAPI Drivers wishing to use ATA power management commands must poll DRDY and, if it is not set, they must also look at the Cylinder registers for the ATAPI signature. If the signature is

present, the ATAPI Driver must issue the ATAPI Device an ATAPI command, re-enabling DRDY, before it can issue an ATA Power management command. Operating systems wishing to use a common ATA power management Driver must also be changed to perform this detection and recovery sequence, if they intend to power-manage ATAPI Devices.

2.14.3. Special Handling of ATA Read and Identify Drive Commands

In order to provide ATAPI Drivers with the ability to force a Device to initialize its ATAPI signature (Cylinder High = EBh, Cylinder Low = 14h) without issuing an SRST, ATAPI Devices shall abort the ATA Read and Identify Drive commands and initialize the task file with the ATAPI signature before clearing BSY.

2.14.4. ATAPI aware BIOS and Driver Considerations

Pre-ATAPI BIOS will not detect or configure ATAPI Devices. Some of the BIOS is capable of configuring ATA hard disks for ATA Mode 3 PIO & IORDY operation. This places a special burden on ATAPI Drivers to detect the presence of any ATA disk drive sharing the same port address and configure the ATAPI Device for a compatible mode of operation.

Note that a special IDE port configuration Driver must be provided by the IDE card manufacturer to configure the cards proprietary IDE configuration control registers. These proprietary IDE card Drivers should be loaded before the ATAPI Driver.

During ATAPI Device detection, ATAPI Device Drivers or ATAPI-aware BIOS should verify that Status=00h (Not BSY, Not RDY) and that the ATAPI signature Cylinder High = EBh, Cylinder Low = 14h are present. If an ATAPI Device is detected, then issue an ATAPI Identify Command to complete the ATAPI detection protocol and re-enable the task file (DRDY=1). If the Device is ready to accept an ATA command but no ATAPI signature is detected, then issue an ATA Read or Identify Drive command to the Device to force the ATAPI Device to initialize it’s signature. Then wait for BSY=0 and re-verify the presence of the ATAPI signature. If there is still no ATAPI signature present, do not configure the Device.

ATAPI-aware BIOS and Drivers should give special attention to managing configurations where ATAPI Drivers share an IDE port address (Cable) with ATA IDE Drives and their Drivers. ATA IDE Drivers frequently issue SRSTs to manage errors thereby causing ATAPI Devices to clear DRDY as part of their SRST ATAPI signature initialization sequence. If the ATAPI Driver already knows that the Device it wishes to issue an ATAPI command to is an ATAPI Device, then it need not take special action since issuing any of the ATAPI commands that do not require DRDY=1, will restore the ATAPI Device’s ability to accept ATA commands. If, however, the ATAPI Driver wishes to issue an ATA command to an ATAPI Device that has received an SRST from an ATA IDE Driver, it should issue the ATAPI Device an ATAPI packet command or Identify to restore the ATAPI Device’s ability to accept ATA commands.

Note that “Newer” BIOS detects the presence of a Drive (see 1.3 ATA Compatibility on page 1) by using the Identify Drive command, but older BIOS use configuration information from outside the IDE/ATA interface. It has also been discovered that very old BIOS may issue an ATA Read command to detect the presence of an ATA IDE Drive. Therefore, the ATA Read and Identify Drive commands shall be aborted by ATAPI Devices. It has also been discovered that some BIOS looks at the status register to detect the presence of an ATA Drive.

Implementor's Note: *Implementors of ATAPI drivers which are intended to share a single cable with a disk and disk driver should ensure that the device has completed any issued commands prior to changing the DRV bit.*

2.14.5. Default Timing

ATAPI Devices compatible with this specification shall support ATA mode 3 timing without requiring the Host system to configure the ATAPI Device using any set features commands. ATAPI Devices must therefore either be fast enough to always supply data at the maximum rate allowed by Mode 3 or the ATAPI Device must be shipped with IORDY enabled.

ATAPI Devices shall revert to their default interface configuration on a Power On Reset or a Hardware Reset.

[**Implmentor's Note:** A low speed drive, Mode 0-2, may affect system performance when sharing the same cable with hard disk drives capable of Mode 3 or faster data transfer timing.]

2.15. ATAPI Command Protocol and DSC Handling

2.15.1. ATAPI Media Access Commands and DSC

All ATAPI media access commands return a completion status with DSC set to 0 upon validation of the packet command and transfer of any associated data. When the drive completes, it will set DSC to 1 to indicate completion. This frees up the IDE bus so the Host can send commands to another IDE device on the same cable. After DSC is set to 1, the Host can send a Request Sense to check for any errors that might have occurred.

If another media access command is issued while DSC is 0, the device will go BSY until the previous command is complete.

If an error condition exists upon receipt of a new command, the new command will be aborted and a deferred error will be reported.

2.15.2. ATAPI Non-media Access Commands and DSC

ATAPI non-media commands and ATA commands (other than reset) are executed immediately, and any error status (if needed) is set, before returning completion status. DSC is unaffected by these commands.

2.15.3. ATAPI Read and Write Commands and DSC

ATAPI Read and Write commands use DSC to indicate buffer availability instead of command completion. An initial Read or Write of 0 blocks is used to switch DSC handshake from completion mode to buffer available mode. Any media access command other than read or write will switch DSC back to completion mode.

Once DSC is in buffer available mode, DSC set to 1 indicates that a specified amount of data can be transferred to/from a drive's buffer without the overhead of tape I/O. The amount that can be transferred is specified in the Continuous Transfer field in mode sense page 2Ah.

3. ATAPI Transport Mechanism

The Transport Mechanism provides for the hardware support to connect the Host Computer to the Peripheral. There will ultimately be two “Models” of operation specified, the Compatibility and Extended Capability Models. The Compatibility Model will look exactly like an ATA/IDE Drive to all existing Drivers, Operating Systems and Hardware. It will operate under many restrictions and will be required to support extra capabilities. The Extended Capability Model will be used to address issues, such as IRQ sharing and multi-threading of commands. The Extended Capability Model will be defined at a later time and will not be defined in this specification. The Compatibility Model is described in the remainder of this section.

3.1. Reset Conditions

There are three types of Reset Condition to which ATAPI Devices shall respond:

- **Power On Reset or Hardware Reset:** the Device executes a series of electrical circuitry diagnostics and sets default values, as well as executing the Master Slave Diagnostic Protocol.
- **ATAPI Soft Reset:** ATAPI Devices shall reset the interface circuitry according to the Set Features requirement upon receipt of the ATAPI Soft Reset Command.
- **ATA SRST:** ATAPI Devices shall provide the normal ATA PDIAG/ DASP sequence and initialize the task file with the ATAPI signature upon detection of SRST. No actual reset of the ATAPI Device shall occur and the DSC bit in the Status Register shall be cleared if any commands are active. Active commands shall not be aborted or stopped.

The Reset Conditions above are listed in order of precedence. That is, Power On or Hardware Reset shall take precedence over ATAPI Soft Reset, which shall take precedence over ATA SRST, which shall take precedence over all other conditions.

3.1.1. Power On or Hardware Reset

Each ATAPI Device, as it is powered on, shall perform appropriate internal reset operations, and internal test operations.

ATAPI Devices upon detection of reset, shall:

1. Clear all Commands and I/O operations in progress.
2. Return to Device's default configuration.
3. Perform the DASP/PDIAG sequence.
4. Return any ATAPI Device operating modes to their appropriate initial conditions, similar to those conditions that would be found after a normal power-on reset. MODE SELECT conditions shall be restored to their last saved values if saved values have been established. MODE SELECT conditions for which no values have been saved shall be returned to their default values.
5. Initialize the Task File Registers as follows: Status = 00h, Error = 01h, Sector Count = 01h, Sector Number = 01h, Cylinder Low = 14h, Cylinder High =EBh and Drive/Head = 00h. A value other than 00 in the status register prior to the receipt of the first ATAPI Command Packet from the Host may cause the ATAPI Device to be incorrectly identified by the Host as an ATA compatible disk Drive. BSY = 0, following any Reset, indicates to the Host that the registers within the Task File have been initialized.

3.2. ATAPI Soft Reset Command and Protocol

ATA specifies a mandatory software reset capability because it provides a recovery mechanism from a class of errors/problems that are recoverable in no other way. The current Drivers invoke this feature at some point in their error recovery procedures today.

The ATA software reset mechanism, SRST, (bit 2 in the Drive Control Register) cannot be used for ATAPI Devices, because resets issued by the ATAPI Driver would also reset any attached hard disk and vice versa.

For a software reset to be useful it must be able to bring the Device's microprocessor back from a busy or hung condition to allow issuance of a diagnostic or some other command. Since the microprocessor is the destination of the reset, we cannot depend on it as part of the reset path. Therefore, ATAPI Soft Reset shall be detected/decoded by the interface controller circuitry and be routed back to the microprocessor as a hardware signal. The Device shall ensure the format integrity of any ongoing write operation before resetting.

Upon detection of the ATAPI Soft Reset command, the Device shall:

1. Set BSY when the command is decoded. When the reset sequence in the Device is complete the Busy status shall be cleared. This will be the only status returned to the Host by the ATAPI Soft Reset command.
2. Initialize the task file with the same information as after a Power On Reset. See section 3.1.1 Power On or Hardware Reset on page 16 for a description of the initialization sequence, except for the DRV bit that shall remain unchanged.

3.3. ATAPI Implementation of ATA SRST

The ATA software reset mechanism, SRST, (bit 2 in the Drive Control Register) cannot be used for ATAPI Devices, because resets issued by the ATAPI Driver would also reset any attached hard disk and vice versa. To solve this ATAPI defines an ATAPI Soft Reset command using a reserved ATA opcode that could be decoded by the interface controller hardware.

To maintain Master/Slave compatibility with ATA disk Drives and prevent detection of ATAPI Devices by non ATAPI-aware BIOS, ATAPI Devices shall implement the following upon receipt of an ATA SRST:

1. Follow the SRST Sequence defined in section 3.3.1 SRST Sequence on page 17, and not the sequences shown in the ATA Specification.
2. Initialize the task file with Status = 00h, Error = 01h, Sector Count = 01h, Sector Number = 01h, Cylinder Low = 14h, Cylinder High =EBh and Drive/Head = 00h.
3. Use the DSC bit to indicate completion of any command executing when SRST was detected.
4. Continue executing commands or play operations.
5. Leave Mode settings or Set Feature settings unchanged.
6. If an ATAPI Device detects SRST while DRQ=1 and the ATAPI Device is selected, then and only then, shall an ATAPI Device abort a command in progress on receipt of an SRST.

3.3.1. SRST Sequence

This sequence does not match the ATA Standard but is used for ATAPI Devices.

1. Host Sets and then clears SRST.
2. Master Device sets Busy, clears DASP & PDIAG and waits up to 3ms for Slave to assert DASP.
3. Slave Device also sets Busy, clears DASP & PDIAG and within 1ms asserts DASP.
4. Master Device waits up to 31s for PDIAG to be asserted.
5. Slave Device deasserts Busy and asserts PDIAG within 30s.
6. Master Device clears Busy.
7. Slave Device deasserts DASP after next command or after 31s.

3.4. Physical Connection

The ATAPI Devices are selected by the Address field in the Drive Select Register. When a single Device is attached to the interface it shall be set as Device 0. When the ATAPI Device is attached along with an ATA Mass Storage Drive, the ATAPI Device will be set as Device 1 and respond as a Slave.

Table 3-1 Preferred Device Connection

Primary Cable		Secondary Cable		
Master	Slave[1]	Master	Slave	Notes
ATA Drive				Normal single cable
ATAPI Device				Normal single cable
ATA Drive	ATAPI Device			Single cable with Streaming Tape
ATAPI Device	ATAPI Device			Two ATAPI Devices on Same Cable.
ATA Drive	ATA Drive			Two Drives on one cable
ATA Drive		ATAPI Device		One Drive and Streaming Tape with two cables
ATA Drive	ATA Drive	ATAPI Device	ATAPI Device	Two Drives and two ATAPI Devices
ATA Drive		ATA Drive	ATAPI Device	Two Drives and Streaming Tape with two cables
ATA Drive	ATA Drive	ATA Drive	ATAPI Device	Three Drives and Tape Device with two cables

3.5. Register Mapping

Communication to or from the Device is through I/O Registers that route the input or output data to or from registers (selected) by a code on signals from the Host (CS1FX-, CS3FX-, DA2, DA1, DA0, DIOR- and DIOW-).

[1]The ATAPI Device ***shall*** be capable of either Master or Slave operation.

3.6. ATAPI Register Map (Packet Command)

Logic conventions are: A = signal asserted, N = signal negated, x = does not matter

Table 3-2 I/O Port Functions/Selection Addresses (Compatibility Model)

Addresses					Functions	
CS1FX	CS3FX	DA2	DA1	DA0	Read (DIOR-)	Write (DIOW-)
					Control Block Registers	
N	A	0	0	0	Floppy A Status	Unused
N	A	0	0	1	Floppy B Status	Unused
N	A	0	1	0	Unused	Floppy Digital Output Register
N	A	0	1	1	Floppy ID / Tape Control	RESERVED
N	A	1	0	0	Floppy Controller Status	RESERVED
N	A	1	0	1	Floppy Data Register	
N	A	1	1	0	Alternate ATAPI Status	Drive Control
N	A	1	1	1	Diskette Change/Drive Address	Not Used
					Command Block Registers	
A	N	0	0	0	Data	
A	N	0	0	1	ATAPI Error Register	ATAPI Features
A	N	0	1	0	ATAPI Interrupt Reason Register	
A	N	0	1	1	Reserved For SAM TAG Byte	
A	N	1	0	0	ATAPI Byte Count Register (bits 0-7)	
A	N	1	0	1	ATAPI Byte Count Register (bits 8-15)	
A	N	1	1	0	Drive Select	
A	N	1	1	1	ATAPI Status	ATA Command

Except for the Data Register, all the ATAPI registers are referenced using Byte (8 Bit) Read and Writes. The Data Register is ALWAYS referenced as a 16 bit word.

The following table is shown for reference only. No word references to the registers (Other than the data register) are allowed by this specification.

Table 3-3 Word Mapping of Task File Registers

Typical Addr	Functions			
	Read (DIOR-)		Write (DIOW-)	
	Control Block Registers			
3x0	Floppy B Status	Floppy A Status	Unused	Unused
3x2	Floppy ID / Tape Control	Unused	Reserved	Floppy Digital Output
3x4	Floppy Data Register	Floppy Controller Status	Floppy Data Register	Reserved
3x6	Change/Drive Address	Alternate ATAPI Status	Unused	ATAPI Device Control
	Command Block Registers			
1x0	Data Register		Data Register	
1	ATAPI Error Register		ATAPI Feature Register	
1x2	Reserved for SAM Tag	ATAPI Interrupt Reason	Reserved for SAM Tag	Sector Count
1x4	ATAPI Byte Count Register		ATAPI Byte Count Register	
1x6	ATAPI Status	Drive Select Register	ATA Command Register	Drive Select Register

D7	D6	D5	D4	D3	D2	D1	D0	
BSY	DRDY	Reserved	DSC	DRQ	CORR	IDX	CHECK	Read

Figure 3-1 ATAPI Status Register (ATA Status Register)

DRDY, DSC, CORR and CHECK shall be valid at Completion Status of the command.

Bit 7	BSY	Busy is set whenever the Device has access to the Command Block.
Bit 6	DRDY	Indicates the Device is capable of responding to an ATA command.
Bit 5	Reserved	
Bit 4	DSC	Cleared during the actual execution of Media Access commands. Devices shall hold BSY if ATAPI commands received while this bit is cleared. While no unrecovered error is detected, this bit cleared may also indicate buffer memory not ready to transfer up to Continuous Transfer Limit. The ready state of the buffer memory defaults to a write operation unless at an implied read position.
Bit 3	DRQ	Data Request - Indicates that the Device is ready to transfer a word or byte of data between the Host and the Device. Information in the ATAPI Interrupt Reason shall also be valid during a Packet Command when the DRQ is set.
Bit 2	CORR	Indicates a Correctable Error occurred within data transferred.
Bit 1	IDX	Reserved for future logical interrupt signal to identify source.
Bit 0	CHECK	Indicates that an error occurred during execution of the previous command. The Error Register contains the Sense Key and Code bits.

D7	D6	D5	D4	D3	D2	D1	D0	
Sense Key				MCR	ABRT	EOM	ILI	Read

Figure 3-2 ATAPI Error Register (ATA Error Register)

Bits 7-4	Sense Key	The Sense Key is defined in Table 6-66 on page 89.
Bit 3	MCR	Media Change Requested is used and defined as in the ATA Standard.
Bit 2	ABRT	Aborted Command is used and defined as in the ATA Standard.
Bit 1	EOM	End Of Media Detected.
Bit 0	ILI	Illegal Length Indication.

D7	D6	D5	D4	D3	D2	D1	D0	
Reserved	Reserved for Tag Type			Reserved			DMA	Write

Figure 3-3 ATAPI Feature Register (ATA Feature Register)

Bit 7	Reserved

Bit 6,5,4	Reserved	Reserved for Tag Type.
Bit 3,2,1	Reserved	Reserved for future enhancement.
Bit 0	DMA (Optional)	Any data for the Command shall be transferred via the DMA interface. Note this does not apply for the Command Packet.

D7	D6	D5	D4	D3	D2	D1	D0	
Byte Count (Bits 0-7)								R/W
Byte Count (Bits (8-15)								R/W

Figure 3-4 ATAPI Byte Count Register (ATA Cylinder High/Low Register)

The Byte Count is used for PIO only; it is ignored for DMA operations.

Prior to the issuance of the Packet Command, the count shall be set to the maximum transfer size per DRQ (at least 512 bytes).

When any data is to be transferred, the ATAPI device shall set the Byte Count to the amount of data that the Host shall transfer and then issue a DRQ interrupt. The contents of this register shall not change during the DRQ.

Implementor's Note: *For commands which transfer less than 512 bytes, such as Inquiry or Mode Sense, the Packet contains an allocation length, which may be used to limit the amount of data transferred to the size of the buffer available in the Host.*

D7	D6	D5	D4	D3	D2	D1	D0	
Reserved						IO	CoD	Read

Figure 3-5 ATAPI Interrupt Reason Register (ATA Sector Count Register)

Bit 0	CoD	Command or Data. When this bit is zero then the information being transferred is user data, when one then the data is Command.
Bit 1	IO	Direction for the Information transfer, where in to the Host is indicated by a value of one and out to the Device is zero.

IO	DRQ	CoD	
0	1	1	Command - Ready to Accept Command Packet Bytes
1	1	1	Message (Future) - Ready to Send Message data to Host
1	1	0	Data To Host- Send command parameter data (e.g. READ)
0	1	0	Data From Host - Receive command parameter data (e.g. WRITE)
1	0	1	Status - Register contains Completion Status

D7	D6	D5	D4	D3	D2	D1	D0	
1	Reserved	1	DRV	SAM LUN				R/W

Figure 3-6 ATAPI Drive Select Register (ATA Drive / Head Select Register)

Bit 3-0	SAM LUN	This field replaces the SCSI LUN within the CDB. This field shall be set to zero.
Bit 4	DRV	This bit selects either Device 0 (DRV=0) or 1 (DRV=1).

D7	D6	D5	D4	D3	D2	D1	D0	
Reserved				1	SRST	nIEN	0	Write

Figure 3-7 ATAPI Device Control Register (ATA Drive Control Register)

Bit 2	SRST	This bit is the Software Reset. The ATAPI Device shall follow the reset sequence for SRST defined in 3.3 ATAPI Implementation of ATA SRST on page 17. There is also a new reset capability for ATAPI Devices utilizing a Soft Reset command (see 3.2 ATAPI Soft Reset Command and Protocol on page 17).
Bit 1	nIEN	This bit enables/disables the interrupt to the Host. When nIEN=0 and the Device is selected, INTRQ shall be enabled through a tri-state buffer. When nIEN=1 or the Device is not selected, the INTRQ signal shall be in a high impedance state

4. Streaming Tape (Task File) Commands

4.1. ATA (Task File) Command Implementation Requirements

This section details the Commands that the Device shall support from the ATA Standard definition of the commands. All ATAPI commands other than “A0” packet command shall use ATA command protocol as defined in the ATA Standard.

4.1.1. ATAPI Soft Reset

See 3.2 ATAPI Soft Reset Command and Protocol on page 17.

4.1.2. Check Power Mode

This command checks the power mode.

If the Device is in, going to, or recovering from the Standby Mode, the Device shall set BSY, set the Sector Count Register to 00h, Clear BSY, and generate an interrupt.

If the Device is in Idle Mode, the Device shall set BSY, set the Sector Count Register to FFh, clear BSY, and generate an Interrupt.

4.1.3. Execute Drive diagnostics

This command shall perform the internal diagnostic tests implemented by the Device. The DRV bit is ignored. Both Devices, if present, shall execute this command. See the ATA Standard (X3T9.2/791D) for more information.

Implementor's Note: *ATAPI Device Drivers issuing the Execute Diagnostics command will cause all ATA and ATAPI Devices to execute a diagnostic command resulting in a Device reset. To prevent unwanted resets and or Driver compatibility issues, ATAPI Drivers should not issue the Execute Diagnostics command. The command is implemented by ATAPI Devices for ATA compatibility only.*

4.1.4. Idle Immediate

This command causes the Device to set BSY, enter the Idle Mode, clear BSY, and generate an interrupt. The interrupt is generated even though the Device may not have fully transitioned to Idle Mode.

4.1.5. Nop

This command enables a Host that can only perform 16-bit register accesses to check Device status. The Device shall respond as it does to an unrecognized command by setting Abort in the Error Register, Error in the Status Register, clearing Busy in the Status Register, and asserting INTRQ.

4.1.6. Packet Command

See section 2.2 PACKET Command on page 5.

4.1.7. ATAPI Identify Device

The ATAPI IDENTIFY DEVICE command enables the Host to receive parameter information from the Device. The parameter words in the buffer have the arrangement and meaning defined in the table below. All reserved bits or words shall be zero. Although many of the fields of information returned by the ATAPI Identify Device command are inappropriate for a Device, the fields that shall be returned with meaningful information are shown in Table 4-1 Identify Device, Fields Supported by Streaming Tape.

Like all ATAPI commands (A0h, A1h and 08h), this command can be issued regardless of the state of the DRDY.

Table 4-1 Identify Device, Fields Supported by Streaming Tape

Word	Description	Used
0	General Configuration	Mandatory
1	Cylinders	No
2	Reserved	No
3	Heads	No
4	Number of unformatted bytes per track	No
5	Number of unformatted bytes per sector	No
6	Number of sectors per track	No
7-9	Reserved	No
10-19	Serial Number	Optional
20	Buffer Type	Optional
21	Buffer Size	Optional
22	ECC bytes available	No
23-26	Firmware revision	Mandatory
27-46	Model Number	Mandatory
47	Multiple Sector Command, Sector Count	No
48	Double Word I/O (shall be 00h for Streaming Tape)	No
49	Capabilities: LBA bit shall be supported; DMA, IORDY bits are optional.	Mandatory
50	Reserved	No
51	PIO Cycle Timing	Mandatory
52	DMA Cycle Timing	Mandatory
53	Validity of words 54-58 and 64-70 in this table	Mandatory
54 - 56	Current Cylinder/Heads/Sectors	No
57-58	Current Capacity	No
59	Reserved	No
60-61	User Addressable Sectors	No
62	Singleword DMA mode	Mandatory
63	Multiword DMA mode	Mandatory
64	Enhanced PIO mode	Mandatory
65	Blind PIO minimum cycle time	Mandatory
66	Recommended Multi Word DMA Transfer Cycle Time	Optional
67	Minimum PIO Transfer Cycle Time without Flow Control	Optional
68	Minimum PIO Transfer Cycle Time with IORDY Flow Control	Optional
69 - 70	Reserved (for advanced PIO support)	No
71 - 127	Reserved	No
128 - 159	Vendor Unique	No
160 - 255	Reserved	No

“Optional” Identify Device words, which are not supported, shall be set to zero.

The ATAPI Identify Device command shall not delay the transfer of the Identify Device data by more than 200 ms after receipt of the command. When the command is issued, the device sets BSY, sets up parameter information in the data buffer, and then sets DRQ and INTRQ.

***Implementor's Note**: All ASCII fields will be byte swapped to maintain compatibility with ATA. For example, "This Example" would be written as "hTsiE axpmel".*

4.1.7.1. General Configuration Word (0)

Table 4-2 Identify Device Data - General Configuration (Word 0)

Bit Byte	7/15	6/14	5/13	4/12	3/11	2/10	1/9	0/8
0	Removable	CMD DRQ Type		Reserved			CMD Packet Size (00)	
1	Protocol Type (10b)		Reserved	Device Type (01)				

Bits 15-14	Protocol Type	This field indicates the protocol in use by the Device. 0Xb = ATA 10b = ATAPI 11b = Reserved
Bit 13	Reserved	This is reserved for future enhancement
Bits 12-8	Device Type	This field indicates the Device type. The peripheral types are described in Table 6-9 Peripheral Device Types on page 45.
Bit 7	Removable	Indicates that the Device has removable media.
Bits 6-5	CMD DRQ Type	This field indicates the command packet DRQ type used by this Device. 00b = Microprocessor DRQ: ATAPI Devices reporting Microprocessor DRQ assertion shall assert DRQ within 3ms of receiving the A0h ATAPI Packet Command. ***Implementor's Note:** Devices reporting Microprocessor DRQ require the Device Driver to poll for up to 3ms or accept the granularity of an available timer tick. These issues may result in undesirable system delays when used with multi-threaded OS Drivers.* 01b = Interrupt DRQ: ATAPI Devices reporting Interrupt DRQ assertion shall assert INTRQ in conjunction with the assertion of the command packet DRQ. These Devices shall assert DRQ within 10ms of receiving the A0h ATAPI Packet Command. 10b = Accelerated DRQ: ATAPI Devices reporting Accelerated DRQ assertion shall assert DRQ within 50us of receiving the A0h ATAPI Packet Command. 11b = Reserved for future use.
Bits 4-2	Reserved	This field is unique for each protocol and is reserved for future use.

Bits 1-0	Command Packet Size	This field indicates the size of the command packets used by this Device. 00b = 12 bytes (ALL Streaming Tape) 01b = 16 bytes (Reserved for future Devices) 1Xb = Reserved for future use.

4.1.7.2. Serial Number (Words 10 - 19)

This optional field shall contain the Device's serial number formatted as right-justified ASCII, padded with spaces (20h). If the field is not supported then it shall be filled with spaces.

4.1.7.3. Buffer Type (Word 20)

The contents of the field are determined by the type of Device. These codes are not typically used by ATAPI Devices, however, for ATAPI Streaming Tape Devices, this field is defined the same as byte 6 and 7 of Capabilities and Mechanical Status Page where byte 6 is bits 15-8 and byte 7 is bits 7-0.

4.1.7.4. Buffer Size (Word 21)

The contents of this field indicate the size of the buffer used for Host transfers and caching by the Device. For the ATAPI Streaming Tape Device this field, multiplied by 512 bytes, defines the size of the write buffer.

4.1.7.5. Firmware Revision (Words 23-26)

The contents of this field are vendor-specific, left-justified, and padded with spaces.

4.1.7.6. Model Number (Words 27 - 46)

The contents of this field are vendor-specific, left-justified, and padded with spaces.

4.1.7.7. Capabilities Word (Word 49)

Table 4-3 Identify Device Data - Capabilities Word (49)

Bit Byte	7/15	6/14	5/13	4/12	3/11	2/10	1/9	0/8
0	Vendor Unique							
1	Reserved	Reserved for Future ATAPI Standard	Reserved for Future ATAPI Standard	Reserved for Pseudo DMA Support	IORDY Supported	IORDY can be disabled	LBA Supported	DMA Supported

Bit 8	DMA Supported	This bit indicates that the Device supports the DMA mode of data transfer.
Bit 9	LBA Supported	Indicates that the Device supports the LBA form of addressing. The ATAPI Devices shall set this bit to one.

Bit 10	IORDY can be disabled	Is used to indicate a Device's ability to enable or disable the use of IORDY. If this bit is set to one, then the Device supports the disabling of IORDY.
Bit 11	IORDY Supported	This is used to help determine whether a Device supports IORDY. If this bit is set to one, then the Device supports IORDY operation. If this bit is zero then the Device may support IORDY (this ensures backward compatibility.)
Bit 13	ATAPI Reserved	Reserved for a Future ATAPI enhancement.
Bit 14	ATAPI Reserved	Reserved for a Future ATAPI enhancement.

4.1.7.8. PIO Data Transfer Cycle Timing (Word 51)

The PIO transfer timing for each ATA Drive falls into categories that have unique parametric timing specifications. To determine the proper Device timing category, compare the contents of this field with the Cycle Time specified in Figure 6 of the ATA document in Appendix B. The value returned in Bits 15-8 shall fall into one of the categories specified, and if it does not, then Mode 0 shall be used to serve as the default timing.

4.1.7.9. DMA Data Transfer Cycle Timing (Word 52)

The DMA transfer timing for each ATA Drive falls into categories that have unique parametric timing specifications. To determine the proper Device timing category, compare the contents of this field with the Cycle Time specified in Figures 8 and 9 of the ATA document in Appendix B. The value returned in Bits 15-8 shall fall into one of the categories specified, and if it does not, then Mode 0 shall be used to serve as the default timing.

4.1.7.10. Field Validity (Word 53)

Bit 0 = 0.

Bit 1 = 1; this bit guarantees that the fields contained in words 64-70 are valid.

4.1.7.11. Single Word DMA Transfer (Word 62)

The low order byte identifies by bit all of the modes that are supported, e.g., if Mode 0 is supported, bit 0 is set. The high order byte contains a single bit set to indicate which mode is active, e.g., if Mode 0 is active, bit 8 is set.

4.1.7.12. Multi Word DMA Transfer (Word 63)

The low order byte identifies by bit all of the modes that are supported, e.g., if Mode 0 is supported, bit 0 is set. The high order byte contains a single bit set to indicate which mode is active, e.g., if Word 0 is active, bit 8 is set.

4.1.7.13. Enhanced PIO Mode (Word 64)

Bits 7 through 0 of the word 64 of the Identify Device parameter information are defined as the Advanced PIO Data Transfer Supported Field. This field is bit significant. Any number of bits may

be set in this field by the Device to indicate which Advanced PIO Modes that it is capable of supporting. Of these bits, bits 7 through 1 are reserved for future advanced PIO modes. Bit 0, if set, indicates that the Device supports PIO Mode 3.

4.1.7.14. Minimum Multi-word DMA Transfer Cycle Time per Word (Word 65)

Word 65 of the parameter information of the IDENTIFY DEVICE command is defined as the Minimum Multi-word DMA Transfer Cycle Time Per Word. This field defines, in nanoseconds, the minimum cycle time that the Device can support when performing Multi-word DMA transfers on a per word basis.

Any Device that supports Multi-word DMA Mode 1 or above shall support this field, and the value in word 65 shall not be less than 150.

If the Device does not support this field, the Device shall return a value of zero in this field.

4.1.7.15. Manufacturer's Recommended Multi-word DMA Transfer Cycle Time (Word 66)

Word 66 of the parameter information of the IDENTIFY DEVICE command is defined as the Manufacturer's Recommended Multi-word DMA Transfer Cycle Time. This field defines, in nanoseconds, the minimum cycle time per word during a single sector Host transfer while performing a multiple sector READ DMA or WRITE DMA commands over all locations on the media under nominal conditions. A cycle time less than this value may cause DMARQ to be deasserted at a rate that may reduce throughput without data corruption.

Any Device that supports Multi-word DMA Mode 1 or above shall support this field, and the value in word 66 shall not be less than the value in word 65.

If the Device does not support this field, the Device shall return a value of zero in this field.

4.1.7.16. Minimum PIO Transfer Cycle Time Without Flow Control (Word 67)

Word 67 of the parameter information of the IDENTIFY DEVICE command is defined as the Minimum PIO Transfer Without Flow Control Cycle Time. This field defines, in nanoseconds, the minimum cycle time that, if used by the Host, the Device guarantees data integrity during the transfer without utilization of flow control.

Any Device that supports PIO Mode 3 or above shall support this field, and the value in word 67 shall not be less than 180.

If the Device does not support this field, the Device shall return a value of zero in this field.

4.1.7.17. Minimum PIO Transfer Cycle Time with IORDY Flow Control (Word 68)

Word 68 of the parameter information of the IDENTIFY DEVICE command is defined as the Minimum PIO Transfer With IORDY Flow Control Cycle Time. This field defines, in nanoseconds, the minimum cycle time that the Device can support while performing data transfers while utilizing IORDY flow control.

Any Device that supports PIO Mode 3 or above shall support this field, and the value in word 68 shall not be less than 180. If the Device does not support this field, the Device shall return a value of zero in this field.

4.1.8. Set Features

The Set Features command is used to set some interface timing and protocol modes. These modes are set at initialization by many BIOSes. The content of the ATAPI Features Register indicates the function to be performed.

Table 4-4 Contents of the Feature Register for Set Features Command

Bit Byte	7	6	5	4	3	2	1	0
0	Feature Set (1) Clear (0)	Feature Number						

Table 4-5 Set Feature Register Definitions

Feature Number	Set Feature Commands	Supported
01h	Enable 8-bit data transfers	No
02h	Enable write cache	No
03h	Set transfer mode based on value in sector count register	Mandatory
33h	Disable retry	No
44h	Vendor unique length of ECC on read long/write long commands	No
54h	Set cache segments to sector count register value	No
5Dh	Enable Shared Interrupts	No
55h	Disable read look-ahead feature	No
66h	Disable reverting to power on defaults	Mandatory
77h	Disable ECC	No
81h	Disable 8-bit data transfers	No
82h	Disable write cache	No
88h	Enable ECC	No
99h	Enable retries	No
AAh	Enable read look-ahead feature	No
ABh	Set maximum prefetch using sector count register value	No
BBh	4 bytes of ECC apply on read long/write long commands	No
CCh	Enable reverting to power on defaults	Mandatory
DDh	Disable Shared Interrupts	No

If the value in the register is not supported or is invalid, the Device sets an Aborted Command error.

At power on, or after a hardware reset, the default mode is the same as that represented by values greater than 80h.

4.1.8.1. Set Transfer Mode (03h)

The Host can choose the transfer mechanism by Set Transfer Mode and specifying a value in the Sector Count Register. The upper 5 bits define the type of transfer and the low order 3 bits encode the mode value. Since both PIO and DMA settings can be active simultaneously, the Device shall maintain independent transfer mode settings for both PIO and DMA.

Table 4-6 Feature Number Description for Set Feature Command

Feature Number	Data Transfer Mode. Mode contained in Sector Count Register		Default Setting
03h	Value in Sector Count Register	Definition	PIO & DMA Mode 0
	00000 00x	PIO Default Transfer Mode	
	00001 xxx	PIO Flow Control Transfer mode x	
	00010 xxx	Single Word DMA mode x	
	00100 xxx	Multi-Word DMA mode x	
	00011 xxx	Reserved (For Pseudo DMA mode)	

If a Device that supports this specification receives a Set Feature command with a Set Transfer Mode parameter and a Sector count Register value of 00000 000, it shall set its default PIO transfer mode.

If a Device that supports this specification receives a Set Feature command with a Set Transfer Mode parameter and a Sector Count Register value of 00000 001 and the Device supports disabling of IORDY, then the Device shall set its default PIO transfer mode and disable IORDY.

4.1.8.2. Disable Reverting to Power On Defaults (66h)

A setting of 66h allows settings of greater than 80h that may have been modified since power on to remain at the same setting after a software reset. At power on, or after a hardware reset, the default mode is the same as that represented by values greater than 80h.

4.1.8.3. Enable Reverting to Power On Defaults (CCh)

A setting of CCh shall cause the Device to revert back to the default for settings of greater than 80h, which may have been modified since power on, after a software reset.

4.1.9. Sleep

This command is the only way to cause the Device to enter Sleep Mode. The Device is spun down, and when it is stopped, BSY is cleared, an interrupt is generated, and the interface becomes inactive.

The only way to recover from Sleep mode is with a software reset or a hardware reset.

NOTE: The use of hardware reset to recover from Sleep mode may be incompatible with continued operation of the Host system.

A Device shall not power on in Sleep Mode nor remain in Sleep Mode following a reset sequence. If the Device reduced power, the reduce power sequence is not executed.

4.1.10. Standby Immediate

This command causes the Device to enter the Standby Mode. The Device may return the interrupt before the transition to Standby Mode is completed. If the Device has already reduced power, the reduce power sequence is not executed.

For Standby immediate the Device may return the interrupt before the transition to Standby Mode is completed.

5. Streaming Tape Device Model

Streaming Tape Devices optimize their use in storing or retrieving user data sequentially. Since access is sequential over a long medium, position changes typically take a long time, when compared to direct-access Devices with a short medium.

The recording medium is tape cartridges or cassettes of various lengths of a flexible substrate coated with a semi-permanent magnetic material. The recording medium is encapsulated into cartridges containing both a supply peel and a take-up reel. Several American National Standards and QIC (Quarter-Inch Cartridge) standards exist covering the construction of cassettes and cartridges for interchange as well as recording techniques for many of the formats or density combinations.

A complete unit composed of the recording medium and its physical carrier is called a volume. Volumes have an attribute of being mounted or demounted on a suitable transport mechanism.

Mounted is the state of a volume when the Device is physically capable of executing commands that cause the medium to be moved. A volume is demounted when it is being loaded, threaded, unloaded, unthreaded, or when not attached to the Device.

Ready is the state of the Device when medium access and non-medium access commands can be executed. The Device is not ready when no volume is mounted or, from the Host's perspective, whenever all medium access commands report “Check Condition” status and a NOT READY Sense Key. Some Devices may have a separate switch function that places the Device in a not ready state even when a volume is mounted.

The write enabled or write protected state determines when the Host may write information on a volume. This attribute is usually controlled by the user of the volume through manual intervention (e.g., thumbwheel switch).

5.1. Medium Attributes

The recording medium has two physical attributes called **beginning-of-medium** (**BOM**) and **end-of-medium** (**EOM**). **Beginning-of-medium** is at the end of the medium that is attached to the take-up reel. **End-of-medium** is at the end of the medium that is attached to the supply reel. In some cases, the medium is permanently affixed to one or both of the reel hubs.

The entire physical length of medium is not usable for recording data. For most volumes, a length of the medium is reserved before the **beginning-of-medium** and after the **end-of-medium** position. This is done to provide sufficient tape wraps onto the reel hub(s) and to ensure that recording starts in an undamaged section of the medium.

The position on the medium where a pattern of recorded signal may be written by one write component is called a track. A Device may write or read from one or more tracks at a time, depending on the format.

On a new volume, recording of one or more tracks begins after mounting the volume and moving from **beginning-of-medium** toward **end-of-medium**. The number of tracks written at one time is called a track group. For recorded volumes, reading in the forward direction follows the same course of tracks as when writing.

If not all tracks are recorded at the same time, and the Device reverses direction when approaching **end-of-medium** and begins writing on remaining tracks, the recording method is called serpentine. For serpentine Devices that record only one track at a time, each physical track represents one track group.

Some multi-track Devices have only one track group, using a parallel storage format that supports the simultaneous recording of all available tracks.

The serpentine and parallel recording formats define tracks as longitudinal patterns of recorded information.

For most recording formats, an area at **beginning-of-medium** contains a format identification as a tone burst or some other recognizable pattern. User data is not recorded in this area. The format identification is an attribute of a volume used for interchange purposes and is defined in applicable standards.

5.1.1. Early Warning

When writing, the Host needs an indication that it is approaching the end of the permissible recording area. This position, called **early-warning** (**EW**), is typically reported to the Host at a position early enough for the Device to write any buffered data to the medium while still leaving enough room for additional recorded labels or filemarks. Some American National Standards include physical requirements for a marker placed on the medium to be detected by the Device as **early-warning**.

For Devices that implement large data buffers, the **early-warning** position defined by a physical marker may be too close to the end of the recording region to permit emptying the data buffer(s). For these Devices, a logical concept of **early-warning** shall be used to signal the Host at an appropriate location prior to the physical marker.

5.1.2. Partitions

Another attribute of a volume is called a partition. Partitions consist of one or more non-overlapped mini-volumes, each with its own beginning and ending points, occupying a single physical volume. Each partition (x) within a volume has a defined **beginning-of-partition** (**BOP**x), an **early-warning** position (**EW**x), and an **end-of-partition** (**EOP**x).

All volumes have a minimum of one partition called partition zero, the default data partition. For Devices that support only one partition, the **beginning-of-partition** zero (**BOP**0) may be equivalent to the **beginning-of-medium** and the **end-of-partition** zero (**EOP**0) may be equivalent to the **end-of-medium**.

When a volume is mounted, it is logically positioned to beginning of the default data partition (**BOP**0). When a REWIND command is received in any partition (x), the Device positions to the **beginning-of-partition** (**BOP**x).

Partitions on a volume do not need to be recorded in any defined order, nor do all partition numbers in a sequence need to be present on a volume. It is sufficient for a Device to be able to locate a partition, given its code value, or determine that it does not exist on the volume. For interchange, information about which partitions are present on a volume may be stored on the volume in a Device-defined area (possibly unavailable to the Host) or the information may be an intrinsic attribute of the Device implementation.

5.2. Quick File Access

Partitions MUST start on track boundaries at the physical **BOT** end of the tape. Partitions shall be used to support the implementation of QFA (Quick File Access). QFA is a feature that provides support for two partitions on the tape cartridge, a directory partition and a data partition. For this specification, the partitions are defined by the Device.

Implementation of QFA allows the Host to partition a tape into two partitions. Note that the default configuration for a tape is a single partition. Devices supporting QFA shall allocate the two partitions as follows:

Partition	Use
0	Data
1	Directory

The partitions are created when the Host issues a Mode Select command using the Medium Partition Page. If the fixed data partitions bit (FDP) is set to one, partition 1 (the directory track) shall be allocated according to a format-specific allocation..

For defining a partition, the FDP bit in the Medium Partition Page is the only field that is to be used by this specification. The remaining fields apply to additional partitions that are not defined by this specification. QIC QFA Devices do not permit the definition of additional partitions.

5.2.1. Changing Partitions

The Host has one method available for changing partitions.

1. The Host can issue a Locate command with the change partition bit (CP) set to one. The partition byte in the Locate command indicates which partition is to become the active partition.

5.2.2. Automatic Format Detection

As most Streaming Tape Devices are capable of reading and writing many different formats, some formats now include MODE SENSE data within the tape header to identify the different format options. Normally, a MODE SELECT issued prior to a read or write operation would explicitly define the format options. If a MODE SELECT command had not been made since the mounting of the volume or with no volume mounted, the Device shall follow these set rules to select the density and block size that will be used implicitly:

1. If no volume is mounted and a MODE SENSE command is issued, the Device shall report its default block size and its native format in the density code field.
2. If a blank or erased tape cartridge is inserted, and a MODE SENSE command is issued after the load process is complete, the Device shall report the default block size and the highest density supported for the cartridge.

If no MODE SELECT command has been made since the mounting of the volume with recorded data, the Device shall follow these set rules to select the density and block size that will be used implicitly:

1. MODE SENSE command is issued after the load process is complete will reflect MODE data written in the tape header if present.
2. If MODE data is not present, the Device will detect the density from the reference burst and use the first data block found on the tape to determine block length.
3. A MODE SENSE, SPACE to EOD, or READ command issued after the load process is complete implicitly set the density and block size.
4. If no MODE SELECT, MODE SENSE, SPACE to EOD, or READ command is issued prior to a WRITE command, the Device shall write at the settings for a blank tape. The Device will have

already retrieved the MODE data or determined the density and block size but shall not use this information in this case.

5.3. Logical Elements within a Partition

The area between **BOP**x and **EOP**x on a typical recorded volume contains at least two types of Host accessible elements, data blocks and tape marks. These elements are controlled and transferred between the Host and the medium using READ, WRITE, and WRITE FILEMARK commands.

5.3.1. Blocks

A unit of data supplied or requested by the Host is called a logical block. Logical blocks are stored according to the specifications of the format for the volume and may be recorded as one or more physical blocks on the medium. When the physical block and the logical block are not recorded in a one-to-one relationship, it is the responsibility of the Device to perform all blocking, de-blocking, padding, stripping, splitting or rebuilding of the logical data block(s) sent by the Host. If the Device is unable to provide the physical to logical translation, as determined by the Capabilities and Mechanical Status Page, only the SPACE command operates using logical blocks but SPACE may not support the reverse direction. As physical blocks may not be sequential, this requires saved READ POSITION information and LOCATE commands for the Host to navigate the medium.

5.3.2. Filemarks

Filemarks are special recorded elements containing no user data. The filemark format is defined in some American National Standards. Hosts traditionally use filemarks to separate user data from labels and logical groupings of data from each other. Since some format standards do not define an explicit **end-of-data** (**EOD**), Host software has often used conventions with filemarks to represent an **EOD** indication. At least one American National Standard specifically defines filemark use for this purpose.

5.3.3. Gaps

Inter-block gaps, the gaps between blocks and filemarks, are introduced on the medium at the time a block or mark is written without explicit action by the Host. Minimum and maximum lengths for inter-block gaps are defined in some American National Standards.

An erase gap may be a length of erased medium or a recorded pattern not distinguishable as a block or mark. Minimum and maximum lengths for erase gaps are defined in some American National Standards while some Devices may have no implementation of an erase gap. For this specification, gaps are defined by the Device.

5.3.4. Blank

After writing data from **BOP**x, the medium is considered to be a contiguous grouping of blocks, filemarks, and gaps. Certain American National Standards define gap lengths which, if exceeded, are to be considered as having reached blank medium. Depending on the format, this blank medium may be treated as an **end-of-data** indication or an unrecoverable medium error causing an interchange error. Unrecorded volumes (new or erased) may exhibit blank medium characteristics if an attempt is made to read or space the volume before data has been written.

For SCSI standards, the device may have been capable of supporting fixed or variable length blocks. The concept of fixed or variable mode for writing and reading blocks only indicates the method by

which the Host specifies the size of a logical block for transfer and not the method of recording physical blocks on the medium. However, for this specification, Devices only support either fixed-length physical or logical blocks of 512 or 1024 bytes. The length of a logical block is always described in bytes. The length of a physical block may or may not be recorded as an exact byte count, depending on the format but the data shall be in blocks of 512 or 1024 bytes.

5.4. Data Buffering

An ATAPI Device contains a temporary storage area capable of holding one or more logical blocks - a data buffer. A Streaming Tape Device data buffer may include any combination of blocks and filemarks in the process of being written to the medium, or it may contain read-ahead data blocks transferred from the medium. (See 2.15 ATAPI Command Protocol and DSC Handling, page15).

A Device will return “Good” status for read and write operations when all data has been successfully transferred to/from the Host from/to the Device data buffer. The Host should only attempt to send read or write commands when the data buffer is able to immediately transfer the data, indicated by DSC set to 1. Otherwise, the command will wait for data to be transferred between the internal buffer and the medium and prevent the bus from being used on a different device.

The ability of the Device to immediately transfer data is determined by the DSC bit in the Status Register. The number of blocks that can be transferred without causing a command delay is determined by the Capabilities and Mechanical Status Page (2Ah).

A read operation is initiated by a READ command with a zero transfer length to start filling the buffer with read-ahead data without holding the bus. The DSC bit within the Status Register will be set when the number of blocks, as determined by the Capabilities and Mechanical Status Page, are ready to be transferred with a read operation.

A WRITE FILEMARK command will not set DSC until all buffered write blocks and filemarks are transferred to the medium. Upon successful completion of this process, which is called a synchronize operation, no blocks or filemarks remain in the data buffer. A synchronize operation has no effect on a data buffer which contains only read-ahead data.

Should an unrecoverable write error occur, the Device generates an error condition to the current active command. If no command is active, the error shall be reported on the next applicable operation as a deferred error. Any data within the Device's buffer not written will be lost but this is only after a catastrophic write failure. The Read Position command may be used to determine the number of buffered blocks not written including the block in error before the unrecoverable error was encountered (if buffered blocks is optionally implemented in Read Position).

A Device with read-ahead data blocks in the data buffer does not report an unrecovered read error until the data block in error is requested by the Host.

5.5. Recorded Element Descriptors (Block Identifiers)

Some recording formats specify that recorded elements (blocks and filemarks) have identifiers included in the recorded information to help determine write sequence and also to help detect Device positioning errors. The identifier values are unique within a partition and may be unique within a volume.

The use of the term block identifier may imply some arithmetic sequence applied to the assignment of recorded elements. The block identifier assignment algorithm may be defined in an applicable format standard.

For some pre-formatted volumes, the identifiers are associated with physical blocks. In variable-block size implementations, the identifier can be associated with a physical block when the logical block and the physical block have a one-to-one relationship on the medium.

Some formats may carry both physical and logical block identifiers recorded on the medium. When a logical block is split over more than one physical block, or multiple logical blocks are concatenated to form a physical block, the logical block identifier and the physical block identifier are not the same. Filemarks may or may not have recorded identifiers, but if identifiers are used in the format, then each mark is assigned a value even if it is not explicitly recorded.

5.6. Positioning

The READ POSITION and LOCATE commands use four-byte fields to hold these format dependent identifiers. For some implementations, this value may correspond to a real physical location; however, it is sufficient for the Device to map the identifier to a value representing the unique recorded element. With this capability, the READ POSITION command may be used to report a Device-defined block identifier and the Host may use this value with a LOCATE command to position to the same location at some future time (provided the volume has not been rewritten in the interim).

5.6.1. Direction and Position Definitions

For Devices, positioning has the connotation of logically being in, at, before, or after some defined place within a volume. This definition means the position is capable of being repeated under the same circumstances. The orientation of usage for the four words (in, at, before, or after) is in one direction, from **BOP**x toward **EOP**x. All positioning defined below is worded from this perspective.

The forward direction is defined as logically progressing from **BOP**x toward **EOP**x. The reverse direction is defined as logically progressing from **EOP**x toward **BOP**x. In serpentine Devices, the logical forward or reverse direction has an alternating relationship to the physical motion of the medium.

The concept of being “in” some position means not being outside a defined region. The definition allows the position to be on the boundary of a defined region. When a volume is first loaded, the logical position is always at the beginning of the default data partition (**BOP**0). Whenever a volume is mounted and the medium motion is stopped, the position is in some partition. While moving between partitions, there is no stable position.

The concept of being “at” some position indicates being positioned to a logical or physical extremity of a partition. A Streaming Tape Device may be positioned at **beginning-of-medium**, at **BOP**x, at **end-of-data** (**EOD**), at **EOP**x, or at **end-of-medium** (**EOM**), since these are stable positions at extremities of a partition.

The concept of being “before” some position indicates that there is some element (data block, filemark, or other defined point) which may be encountered when moving toward **EOP**x, if the proper commands are issued. Being positioned before a particular data block means that if the Device receives a valid READ command, the data block is transferred to the Host. This position may also be before **Ew**x and **EOP**x, since these are defined points within any partition. However, if data has not been written to the **end-of-partition**, these points may not be accessible by the Host.

The concept of being “after” some position indicates that there is some element (data block, filemark, or other defined point) on the **BOP**x side of the current position that may be encountered if the proper

commands are issued. When a READ command for a single data block has been successfully executed, the logical position is after the transferred data block.

5.7. Error Reporting

If any of the following conditions occurs during the execution of a command, the Device shall return "Check Condition" status. The appropriate Sense Key and Additional Sense Code should be set.

In the case of an unrecovered read or write error, if the read-write error recovery page is implemented, the current values specify the error recovery criteria. If this page is not implemented, the error recovery is vendor specific.

In the case of an unrecovered read error, the Valid bit shall be set to one and the Information field shall be set to the difference (residue) of requested transfer length minus the actual number of blocks read (not including the unrecovered block). Upon termination, the logical position shall be after the unrecovered block.

In the case of an unrecovered write error, the Valid bit shall be set to one and the Information field shall be set to the difference (residue) of the sum of the pending requested transfer lengths minus the actual number of blocks and filemarks written. The value in the Information field may exceed the transfer length for the last command. For formats able to restart a write operation, upon termination, the logical position shall be after the unrecovered block.

6. Streaming Tape Packet Commands

6.1. Packet Command Implementation Requirements

The first byte of all ATAPI Device Command Packets shall contain an operation code as defined in this Specification. ATAPI Devices shall implement all commands with mandatory operation codes.

6.1.1. Reserved

Reserved bits, fields, bytes, and code values are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other specifications. A reserved bit, field, or byte shall be set to zero, or in accordance with a future extension to this specification. The recipient shall not check reserved fields.

6.2. ATAPI Command Packet Description

An ATAPI command is communicated by sending a Command Packet to the ATAPI Device. For several commands, the Command Packet is accompanied by a list of parameters sent upon receiving an interrupt following the Command Packet being sent. See the specific commands for detailed information.

The Command Packet always has an operation code as its first byte.

For all commands, if there is an invalid parameter in the Command Packet, then the ATAPI Device shall abort the command without altering the medium.

Table 6-1 Typical Command Packet for Most Commands

Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Reserved							
2								
3	Reserved							
4	Allocation Length, Parameter List Length, or Mode							
5	(MSB) Parameter Pointer							
6	(LSB)							
7	(MSB) Allocation Length or Parameter List Length							
8	(LSB)							
9	Reserved							
10	Reserved							
11	Reserved							

Table 6-2 Typical Command Packet for Read or Write Operations

Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Reserved							1 (BT)
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

6.2.1. Operation Code

The operation code of the Command Packet has a group code field and a command code field. The three-bit group code field provides for eight groups of command codes. The five-bit command code field provides for thirty-two command codes in each group. Thus, a total of 256 possible operation codes exists. Operation codes are defined in the subsequent sections.

Table 6-3 Operation Code

Bit	7	6	5	4	3	2	1	0
	Group Code			Command Code				

Note that the Group/Command code fields have been kept for backward compatibility and are not used by ATAPI.

6.2.2. Logical Block Address

The logical block address shall begin with block zero and be contiguous up to the last logical block.

6.2.3. Transfer Length

The Transfer Length Field specifies the amount of data to be transferred, usually the number of blocks. For several commands the transfer length indicates the requested number of bytes to be sent as defined in the command description. For these commands the Transfer Length Field may be identified by a different name. See the following descriptions and the individual command descriptions for further information.

In commands that use multiple bytes for the transfer length, a transfer length of zero indicates that no data transfer shall take place. A value of one or greater indicates the number of blocks that shall be transferred.

6.2.4. Parameter List Length

The Parameter List Length is used to specify the number of bytes to be sent to the Device. This field is typically used in Command Packets for parameters that are sent to a Device (e.g. mode

parameters, diagnostic parameters, etc.). A parameter length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

6.2.5. Allocation Length

The Allocation Length Field specifies the maximum number of bytes that the Host computer has allocated for returned data. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The Device shall terminate the data transfer when allocation length bytes have been transferred or when all available data have been transferred to the Host, whichever is less. The allocation length is used to limit the maximum amount of data (e.g. sense data, mode data, etc.) returned to the Host.

6.3. Status

A Status byte shall be sent from the Device to the Host at the completion of each command unless the command is terminated by one of the following events:

1. a reset condition;
2. an unexpected event

Status is normally presented at the end of a command, but in some cases may occur prior to transferring the Command Packet.

For a description of the Status Byte see Figure 3-1 ATAPI Status Register (ATA Status Register) on page 20.

6.4. Command Processing Considerations and Exception Conditions

The following sections describe some exception conditions and errors associated with command processing and the sequencing of commands.

6.4.1. Parameter Rounding

Certain parameters sent to an ATAPI Device with various commands contain a range of values. ATAPI Devices may choose to implement only selected values from this range. When the ATAPI Device receives a value that it does not support, it either rejects the command ("Check Condition" status with ILLEGAL REQUEST Sense Key) or it rounds the value received to a supported value. The ATAPI Device shall reject unsupported values unless rounding is permitted in the description of the parameter.

Rounding of parameter values, when permitted[2], shall be performed as follows - An ATAPI Device that receives a parameter value that is not an exact supported value shall adjust the value to one that it supports and shall return "Check Condition" status with a Sense Key of RECOVERED ERROR. The Additional Sense Code shall be set to ROUNDED PARAMETER. The Host is responsible for issuing an appropriate command to learn what value the ATAPI Device has selected.

[2]. Generally, the ATAPI Streaming Tape Drive should adjust maximum-value fields down to the next lower supported value than the one specified by the Host Computer. Minimum-value fields should be rounded up to the next higher supported value than the one specified by the Host Computer. In some cases, the type of rounding (up or down) is explicitly specified in the description of the parameter.

6.5. Unit Attention Condition

The ATAPI Device shall generate a unit attention whenever the Device has been reset by a hard reset condition, or by a power-on reset. The Device shall also generate a unit attention condition whenever one of the following events occurs:

1. A removable medium may have been changed;
2. The version or level of microcode has been changed;
3. The mode parameters in effect have been restored from non-volatile memory;
4. Any other event occurs that requires the attention of the Host.

The ATAPI Device may queue unit attention conditions. After the first unit attention condition is cleared, another unit attention condition may exist (e.g. a power on condition followed by a microcode change condition).

The unit attention condition shall persist, until the Host clears the condition as described in the following paragraphs.

If an INQUIRY command or any non packet ATA command is received from the Host with a pending unit attention condition, the ATAPI Device shall perform the INQUIRY command and shall **not** clear the unit attention condition.

If a REQUEST SENSE command is received from the Host with a pending unit attention condition, then the ATAPI Device shall:

1. report the unit attention condition, may discard any pending sense data, and clear the unit attention condition.

If the Host issues a command other than INQUIRY or REQUEST SENSE while a unit attention condition exists, the ATAPI Device shall **not** perform the command and shall report “Check Condition” status unless a higher priority status as defined by the ATAPI Device is also pending (e.g. BUSY).

6.6. Commands and Parameters

The ATAPI Streaming Tape commands are derived from the QIC-121 command set.

The interface uses logical rather than physical addressing for all data blocks. Each Device may be interrogated to determine how many blocks it contains.

Commands are classified as mandatory, optional, or vendor-specific. ATAPI Devices are required to implement all mandatory commands and may implement other commands as well. ATAPI Devices contain commands that facilitate the writing of self-configuring software Drivers that can discover all necessary attributes without prior knowledge of specific peripheral characteristics.

6.7. ATAPI Media Access Command Considerations

ATAPI media access commands are a class of commands that normally return good completion status to the Host system before they are finished executing the command. The purpose of this is to allow the Host to execute more than one command at a time on the same IDE cable.

ATAPI Devices use the DSC bit to indicate the completion of commands. No INTRQ is issued by these Devices when the DSC bit is set, so it the responsibility of the ATAPI Driver to poll this bit to determine the completion. See also 2.15 ATAPI Command Protocol and DSC Handling on page 15.

6.8. ATAPI Packet Commands for Streaming Tape Devices

Table 6-4 Packet Commands Supported by Streaming Tape Devices

Command Description	Opcode	Type	DSC Restrictive	Media Access	Reference
ERASE	19h	M	✓	✓	6.8.1 on page 43
INQUIRY	12h	M			6.8.2 on page 44
LOAD/UNLOAD	1Bh	M	✓	✓	6.8.3 on page 47
LOCATE	2Bh	M	✓	✓	6.8.4 on page 49
LOG SELECT	4Ch	M			6.8.5 on page 50
LOG SENSE	4Dh	M			6.8.6 on page 52
MODE SELECT	15h	M			6.8.7 on page 59
MODE SENSE	1Ah	M			6.8.8 on page 61
READ	08h	M	buffer	✓	6.8.9 on page 79
READ POSITION	34h	M			6.8.10 on page 81
REQUEST SENSE	03h	M			6.8.11 on page 84
REWIND (Rezero Unit on CD-ROM)	01h	M	✓	✓	6.8.12 on page 93
SPACE	11h	M	✓	✓	6.8.13 on page 94
TEST UNIT READY	00h	M			6.8.14 on page 96
WRITE	0Ah	M	buffer	✓	6.8.15 on page 97
WRITE FILEMARK	10h	M	✓	✓	6.8.16 on page 99
M = command implementation is mandatory. O = command implementation is optional.					

[**Editor's Note:** Any additional commands/capabilities required by a particular device that are not provided in this specification should be implemented per QIC-121.]

[**Editor's Note:** Many unused bit fields contain fixed values, with parenthetical names that are not defined. (ie 0 (SP). These explicitly indicates which fields from QIC-121 are not supported by this specification.]

6.8.1. Erase Command

The ERASE command causes part or all of the medium to be erased beginning at the current position. As used here, “erased” means either the medium shall be erased or a pattern shall be written on the medium that appears as a gap.

Table 6-5 Erase Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (19h)							
1	Reserved						Reserved (Immed)	1 (Long)
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

Some Devices may reject an ERASE command if not at **beginning-of-partition**.

Table 6-6 Recommended Sense Key, ASC and ASCQ for ERASE Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
03	51	00	ERASE FAILURE (FORMAT FAILURE)
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED
07	27	00	WRITE PROTECTED

6.8.2. INQUIRY Command

The INQUIRY command requests that information regarding parameters of the Device be sent to the Host. An option allows the Host to request additional information about the Device.

Table 6-7 INQUIRY Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (12h)							
1	Reserved							Reserved (EVPD)
2	Reserved (Page Code)							
3	Reserved							
4	Allocation Length							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

The INQUIRY command shall return “Check Condition” status only when the Device cannot return the requested INQUIRY data. The INQUIRY data should be returned even though the Device may not be ready for other commands.

If an INQUIRY command is received with a pending unit attention condition (i.e. before the Device reports “Check Condition” status), the Device shall perform the INQUIRY command and shall not clear the unit attention condition.

6.8.2.1.Standard INQUIRY Data

The standard INQUIRY data contains 36 required bytes, followed by a variable number of vendor-specific parameters. Bytes 56 through 95, if returned, are reserved for future standardization.

Table 6-8 INQUIRY Data Format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved (Peripheral Qualifier)			Peripheral Device Type (01)				
1	RMB	Reserved						
2	ISO Version		ECMA Version			ANSI Version (02)		
3	Reserved (AENC)	Reserved (TrmIOP)	Reserved		Response Data Format (02)			
4	Additional Length (n-4)							
5	Reserved							
6	Reserved							
7	Reserved (RelAdr)	Reserved (WBus32)	Reserved (WBus16)	Reserved (Sync)	Reserved (Linked)	Reserved	Reserved (CmdQue)	Reserved (SftRe)
8-15	Vendor Identification							
16-31	Product Identification							
32-35	Product Revision Level							
36-55	Vendor-specific							
56-95	Reserved							
Reserved (Vendor Specific Parameters)								
96								
n								

The Device-type field identifies the Device. It is defined in Table 6-9 Peripheral Device Types on page 45.

[**Editor's Note:** Due to hardware restrictions all pages should be padded to a multiple of 4 length.]

6.8.2.2. Using the INQUIRY Command

The INQUIRY command may be used by the Host to determine the configuration of the Device. Devices respond with information that includes their type and Specification level and may include the vendor's identification, model number and other useful information.

Table 6-9 Peripheral Device Types

Code	Description
00h	Direct-access Device (e.g. magnetic disk)
01h	Streaming Tape Device (QIC-121 Architectural Model)
02h - 04h	Reserved
05h	CD-ROM Device
06h	Reserved
07h	Optical memory Device (e.g. some optical disks)
08h - 1Eh	Reserved
1Fh	Unknown or no Device type

The Peripheral Device Type shall be set to 01h to indicate a Streaming Tape Device.

A **R**emovable **M**edium **B**it (**RMB**) of zero indicates that the medium is not removable. A **RMB** bit of one indicates that the medium is removable. Streaming Tape Devices should always report "Removable".

The usage of non-zero code values in the **ISO Version** and **ECMA Version** fields are defined by the International Organization for Standardization and the European Computer Manufacturers Association, respectively.

The **ANSI Version** field must contain a 02h to comply with this version of this Specification.

A **Response Data Format** value of 02h indicates that the data shall be in the format specified in this Specification. Other **Response Data Format** values are reserved.

The **Additional Length** field shall specify the length in bytes of the parameters. If the allocation length of the Command Packet is too small to transfer all of the parameters, the additional length shall not be adjusted to reflect the truncation.

ASCII data fields shall contain only graphic codes (i.e. code values 20h through 7Eh). Left-aligned fields shall place any unused bytes at the end of the field (highest offset) and the unused bytes shall be filled with space characters (20h). Right-aligned fields shall place any unused bytes at the start of the field (lowest offset) and the unused bytes shall be filled with space characters (20h).

The **Vendor Identification** field contains 8 bytes of ASCII data identifying the vendor of the product[3]. The data shall be left aligned within this field.

The **Product Identification** field contains 16 bytes of ASCII data as defined by the vendor. The data shall be left-aligned within this field.

The **Product Revision Level** field contains 4 bytes of ASCII data as defined by the vendor. The data shall be left-aligned within this field.

Table 6-10 Recommended Sense Key, ASC and ASCQ for INQUIRY Errors

Sense Key	ASC	ASCQ	Description of Error
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET

[3]. It is intended that this field provide a unique vendor identification of the manufacturer of the ATAPI Streaming Tape Drive. In the absence of a formal registration procedure, QIC maintains a list of vendor identification codes in use. Vendors are requested to voluntarily submit their identification codes to QIC to prevent duplication of codes.

6.8.3. LOAD UNLOAD Command

The LOAD UNLOAD command requests that the Device enable or disable further media access operations. Media access commands are defined in Table 6-4 Packet Commands Supported by Streaming Tape Devices. This command may also be used to request a re-tension function. Prior to performing the load or unload operation, the Device shall ensure that all buffered data, and filemarks have been transferred to the medium.

Table 6-11 Load Unload Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (1Bh)							
1	Reserved							Reserved (Immed)
2	Reserved							
3	Reserved							
4	Reserved					EOT	Re-Ten	Load
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

An **end-of-tape** (**EOT**) bit of one indicates that an unload operation (**Load** bit set to zero) shall position the medium at **end-of-medium** for removal from the Device. An **EOT** bit of zero indicates that an unload operation shall position the medium at **beginning-of-medium** for removal from the Device.

An **EOT** bit of one and a **Load** bit of one shall cause the Device to return “Check Condition” status and the Sense Key shall be set to ILLEGAL REQUEST in the sense data.

A **Re-Ten**sion (**Re-Ten**) bit of one indicates that the medium on the Device shall be correctly tensioned. Implementation of the re-tensioning function is Device specific that typically consists of one end-to-end pass on the cartridge.

If the **Load** bit is set to one, the medium shall be loaded and positioned to the **beginning-of-partition** zero. If the **Load** bit is zero, the medium in the Device shall be positioned for removal at the extreme position along the medium specified by the **EOT** bit. Following successful completion of an unload operation, the Device shall return “Check Condition” status with the Sense Key set to NOT READY for all subsequent medium-access commands until a new volume is mounted or a load operation is successfully completed.

Table 6-12 Load, EOT, and Retension Bit Combinations

Load	EOT	Re-Ten	Meaning
0	0	0	unload
0	0	1	retension then unload at **BOT**
0	1	0	unload at **EOT**
0	1	1	retension then unload at **EOT**
1	0	0	load
1	0	1	load and retension
1	1	0	"Check Condition"
1	1	1	"Check Condition"

Implementor's Note: *Following a hard reset or a cartridge insertion, an implied or automatic load may be performed by the Device.*

Table 6-13 Recommended Sense Key, ASC and ASCQ for LOAD UNLOAD Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
04	53	00	MEDIA LOAD EJECT FAILED
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
05	53	02	MEDIUM REMOVAL PREVENTED
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.4. LOCATE Command

LOCATE command causes the Device to position to the specified block address in a specified partition. Upon completion, the logical position shall be before the specified location. Prior to performing the locate operation, the Device shall ensure that all buffered data, and filemarks have been transferred to the medium.

Table 6-14 Locate Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (2Bh)							
1	Reserved					0 (BT)	CP	Reserved (Immed)
2	Reserved							
3	(MSB) Block Address							
4								
5								
6								(LSB)
7	Reserved							
8	- Partition -							
9	Reserved							
10	Reserved							
11	Reserved							

The **B**lock **T**ype (**BT**) bit of zero indicates the value in the block address field shall be interpreted as a sequential Logical Block Address with block zero originating at the **beginning-of-partition**.

[Implementor's Note: If the **BT** bit is set to one, it will be ignored, and the Block Address field will be interpreted as a Logical Block Address.]

A **C**hange **P**artition (**CP**) bit of one indicates that a change to the partition specified in the partition field is to occur prior to positioning to the block specified in the block address field. A **CP** bit of zero indicates no partition change is to be made and the partition field is to be ignored.

The **Block Address** field specifies the block address to which the Device shall position the medium.

The **Partition** field specifies which partition to select if the **CP** bit is one.

Table 6-15 Recommended Sense Key, ASC and ASCQ for LOCATE Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
04	15	00	RANDOM POSITIONING ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.5. LOG SELECT Command

The LOG SELECT command provides a means to manage statistical information maintained by the Device about the Device. Devices that implement the LOG SELECT command shall also implement the LOG SENSE command. Structures as log parameters within log pages are defined as a way to manage the log data. The LOG SELECT command provides for sending zero or more log pages of data. This specification defines the format of the log pages, but does not define the exact conditions and events that are logged.

Table 6-16 Log Select Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (4Ch)							
1	Reserved						PCR	0 (SP)
2	PC		Reserved					
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)	Parameter List Length						
8								(LSB)
9	Reserved							
10	Reserved							
11	Reserved							

A **P**arameter **C**ode **R**eset (**PCR**) bit of one and a parameter list length of zero shall cause all implemented parameters to be set to zero. If the **PCR** bit is one and a the parameter list length is greater than zero, the command is terminated with “Check Condition” status. The Sense Key shall be set to ILLEGAL REQUEST and the Additional Sense Code shall be set to INVALID FIELD IN COMMAND PACKET. A **PCR** bit of zero specifies that the log parameters shall not be reset.

The **P**age **C**ontrol (**PC**) field defines the type of parameter values to be selected. The page control field is defined in Table 6-17 Page Control Field.

Table 6-17 Page Control Field

PC Value	LOG SENSE	LOG SELECT
00b	Reserved	Reserved
01b	Current Cumulative Values	Cumulative Values
10b	Reserved	Reserved
11b	Reserved	Reserved

[Implementor's Note: LOG SENSE commands should be issued prior to issuing LOG SELECT commands to determine supported pages and page lengths**.]**

Table 6-18 Recommended Sense Key, ASC and ASCQ for LOG SELECT Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
05	1A	00	PARAMETER LIST LENGTH ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
05	26	00	INVALID FIELD IN PARAMETER LIST
05	26	01	PARAMETER NOT SUPPORTED
05	26	02	PARAMETER VALUE INVALID
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.6. LOG SENSE Command

The LOG SENSE command provides a means to retrieve statistical information maintained by the Device about the Device. It is a complementary command to the LOG SELECT command.

Table 6-19 Log Sense Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (4Dh)							
1	Reserved						Reserved (PPC)	Reserved (SP)
2	PC (01)		Page Code					
3	Reserved							
4	Reserved							
5	(MSB) Parameter Pointer							
6								(LSB)
7	(MSB) Allocation Length							
8								(LSB)
9	Reserved							
10	Reserved							
11	Reserved							

The **P**age **C**ontrol (**PC**) field defines the type of parameter values to be selected (see Table 6-17 Page Control Field page 50). The parameter values returned by a LOG SENSE command are determined as follows:

1. The specified parameter values at the last update (in response to a LOG SELECT or LOG SENSE command or done automatically by the Device for cumulative values).
2. The default values if an update has not occurred since the last power-on, hard RESET condition, or ATAPI Soft Reset.

The **Page Code** field identifies which page of data is being requested. If the **Page Code** is reserved or not implemented, the Device shall terminate the command with “Check Condition” status. The Sense Key shall be set to ILLEGAL REQUEST with the Additional Sense Code set to INVALID FIELD IN COMMAND PACKET.

The **Parameter Pointer** field allows requested parameter data to begin from a specific parameter code to the maximum **Allocation Length** or the maximum parameter code supported by the Device, whichever is less. If the value of the **Parameter Pointer** field is larger than the largest available parameter code that can be returned by the Device on the specified page, the Device shall terminate the command with “Check Condition” status. The Sense Key shall be set to ILLEGAL REQUEST and the Additional Sense Code shall be set to INVALID FIELD IN COMMAND PACKET.

Log parameters within the specified log page shall be transferred in ascending order according to parameter code.

Table 6-20 Recommended Sense Key, ASC and ASCQ for LOG SENSE Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
04	15	00	RANDOM POSITIONING ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.6.12.Log Parameters

This section defines the descriptors and pages for log parameters that may be used with Devices.

The log page codes for Devices are defined in Table 6-21.

Table 6-21 Log Page Codes

Page Code	Description	Reference
00h	Supported Log Pages	6.8.6.29 on page 55
01h	Reserved	
02h	Error Counter Page (Write) Page	6.8.6.38 on page 56
03h	Error Counter Page (Read) Page	6.8.6.38 on page 56
04h - 30h	Reserved	
31h	Tape Capacity Page	6.8.6.39 on page 58
32h - 3Eh	Vendor Specific	
3Fh	Reserved	

This section describes the log page structure and the log pages that are applicable to all Devices. Each log page begins with a four-byte page header followed by zero or more variable-length log parameters defined for that page.

Table 6-22 Log Page Format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved		Page Code					
1	Reserved							
2	(MSB)		Page Length (n-3)					
3								(LSB)
	Log Parameter(s)							
4	Log Parameter (First)							
x+3	(Length x)							
n-y	Log Parameter (Last)							
n	(Length y)							

The **Page Code** field identifies which log page is being transferred.

The **Page Length** field specifies the length in bytes of the following log parameters. If the Host sends a **Page Length** that results in the truncation of any parameter, the command shall terminate with "Check Condition" status. The Sense Key shall be set to ILLEGAL REQUEST with the Additional Sense Code set to INVALID FIELD IN PARAMETER LIST.

[**Editor's Note:** Due to hardware restrictions all pages should be padded to a multiple of 4 length.]

Most log pages contain one or more special data structures called log parameters. Log parameters may be data counters that record a count of a particular event (or events) or log parameters may be list parameters (strings) which contain a description of a particular event.

Table 6-23 Log Parameter

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			Parameter Code				
1								(LSB)
2	0 (DU)	1 (DS)	0 (TSD)	0 (ETC)	00 (TMC)		Reserved	0 (LP)
3	Parameter Length (n-3)							
4	Parameter Value							
n								

Each log parameter begins with a four-byte parameter header followed by one or more bytes of **Parameter Value** data.

[**Editor's Note:** Due to hardware restrictions all parameter values should be padded to a multiple of 4.]

The **Parameter Code** field identifies which log parameter is being transferred for that log page.

A **D**isable **S**ave (**DS**) bit of one indicates that the Device does not support saving that log parameter in response to a LOG SELECT command with a SP bit of one.

The **Parameter Length** field specifies the length in bytes of the following parameter value. If the Host sends a **Parameter Length** value that results in the truncation of the **Parameter Value**, the Device shall terminate the command with "Check Condition" status. The Sense Key shall be set to ILLEGAL REQUEST with the Additional Sense Code set to INVALID FIELD IN PARAMETER LIST.

Implementor's Note:

1. For log page codes 02h and 03h, the parameter length field shall be set to 4 and thus the **Parameter Value** shall be 4 bytes long.
2. For log page code 30h, the parameter length field shall be set to 8 and thus the **Parameter Value** shall be 8 bytes long.

6.8.6.29.Supported Log Pages Code 00h

The supported log page returns the list of log pages implemented. Devices that implement the LOG SENSE command shall implement this log page.

Table 6-24 Supported Log Page

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved		Page Code (00h)					
1	Reserved							
2	(MSB)			Page Length (n-3)				
3								(LSB)
4	Supported Page List							
n								

This page is not defined for the LOG SELECT command. This log page returns the list of supported log pages.

The page length field specifies the length in bytes of the following supported page list,

The supported page list field shall contain a list of all log page codes implemented by the Device in ascending order beginning with page code 00h.

6.8.6.38.Error Counter Pages Codes 02, 03

This section defines the optional error counter pages for write errors (page code 02h), and read errors (page code 03h). Table 6-23 defines the page format for these pages. A page can return one or more log parameters which record events defined by the parameter codes.

[Implementor's Note: Devices shall only reset its error counters at power-on or on command from the Host. Table 6-25 defines the parameter codes for the error counter pages. Support of each log parameter is optional.]

Table 6-25 Parameter Codes for Error Counter Pages

Parameter Code	Description
0000h	Errors corrected on-the-fly
0001h	Errors corrected by other means
0002-7FFFh	Reserved
8000h-8003h	Reserved
8004-FFFFh	Vendor specific

6.8.6.38.1.Error Counter Page (Read)

This section defines the parameter codes for the Error Counter Page (Read), page code 03h. Table 6-26 defines the counters and their use.

The following techniques can be used to attempt to recover bad blocks when reading from tape.

1. ECC. An ECC algorithm is applied to the data blocks and parity blocks within a frame in order to correct the bad blocks. This technique can only be used when a minimum number of bad blocks are detected in a frame.
2. Tape re-positions. The tape is re-positioned so that the frame read operation can be repeated.
3. Head adjustments. The head position in relation to the track is slightly modified. Head adjustments are done in combination with tape re-positions.
4. Signal threshold changes. The read amplitude dc values are modified slightly to capture more signal transitions. Signal threshold changes are made in combination with tape re-positions.

Table 6-26 Parameter Codes for Error Counter Page (Read) - Page Code 03h

Code	Description
0000h	Number of blocks recovered by ECC alone.
0001h	Number of blocks recovered without using ECC. This count is incremented when blocks are recovered due to: - tape re-positioning alone - combination of tape re-positioning and head adjustments - combination of tape re-positioning and signal threshold changes. However, no additional ECC processing is required.

Implementor's Note:

1. The term "blocks" refers to physical blocks on tape.
2. A block read error is only counted as being in error on the first attempt to read the block.

6.8.6.38.2.Error Counter Page (Write)

This section defines the parameter codes for the Error Counter Page (Write), page code 02h). Table 6-27 defines the counters and their use.

The following techniques can be used to correct errors detected when writing to tape.

1. Rewrites. If a bad block is detected during the read after write operation, the block will be rewritten until it reads correctly or until the maximum rewrite count is reached. Note that the maximum rewrite count is defined by the physical tape format. Rewrites facilitate tape streaming.
2. Tape re-positioning. Re-positioning during write errors applies only to rewritable data frames.

NOTE: The distance between the write and the read heads may be large compared to the data blocks, therefore, there may be several blocks between the block that is accessed by the write head and the block that is accessed by the read head. This means that for a write retry without re-positioning several blocks may be discarded.

Table 6-27 Parameter Codes for Error Counter Page (Write) - Page Code 02h

Code	Description
0000h	Number of rewrites. Blocks that follow the bad block but precede the last block written will also be re-written, however, these blocks shall not be included in this count.
0002h	Number of re-positions. This count is incremented when tape re-positioning is performed due to an error when attempting to write a re-writable frame. This counter is only used with tape formats which support overwrite.
0003h	Total number of blocks recovered.
0006h	Number of physical blocks not recovered.

NOTES: A frame is in error if any one block, or any number of blocks within the frame cannot be read. The term "blocks" refers to physical blocks on tape.

6.8.6.39.Tape Capacity Page Code 31h

Table 6-28 Tape Capacity Page Code 31h

Bit Byte	7	6	5	4	3	2	1	0
0	Page Code (31h)							
1	Reserved							
2	(MSB) Page Length (32)							
3								(LSB)
4	(MSB) Parameter Code 1h							
5								(LSB)
6	0 (DU)	1 (DS)	0 (TSD)	0 (ETC)	0 (TMC)		Reserved	0 (LP)
7	Parameter Length (4)							
8	(MSB) Remaining Capacity, Partition 0							
11								(LSB)
12	(MSB) Parameter Code 2h							
13								(LSB)
14	0 (DU)	1 (DS)	0 (TSD)	0 (ETC)	0 (TMC)		Reserved	0 (LP)
15	Parameter Length (4)							
16	(MSB) Remaining Capacity, Partition 1							
19								(LSB)
20	(MSB) Parameter Code 3h							
21								(LSB)
22	0 (DU)	1 (DS)	0 (TSD)	0 (ETC)	0 (TMC)		Reserved	0 (LP)
23	Parameter Length (4)							
24	(MSB) Maximum Capacity, Partition 0							
27								(LSB)
28	(MSB) Parameter Code 4h							
29								(LSB)
30	0 (DU)	1 (DS)	0 (TSD)	0 (ETC)	0 (TMC)		Reserved	0 (LP)
31	Parameter Length (4)							
32	(MSB) Maximum Capacity, partition 1							
35								(LSB)

For single partition tapes, **Partition 1** values shall always be zero.

The **Remaining Capacity** for non-current partitions shall be the same as the **Maximum Capacity** for that partition.

The **Maximum Capacity** values are valid only after the load operation is completed. **Remaining Capacity** is valid after a SPACE to **EOD,** WRITE, or WRITE FILEMARK and DSC bit is set. Following LOCATE, REWIND, SPACE Block or Filemark, READ, LOAD and MODE SELECT the **Remaining Capacity** reflects a progress value as a ratio of **Remaining** and **Maximum Capacity**.

Capacities are multiplied by 1024 to determine the number of bytes. These values are estimates and, where compression is implemented, they assume a 1:1 compression ratio and allow for worst-case write retry. Allowances for frame grouping, system log areas, **EOD** and **EOP** areas, and headers are not included in these capacity values to ensure a conservative estimate.

6.8.7. MODE SELECT Command

The MODE SELECT command provides a means to specify medium or Device parameters. Devices that implement the MODE SELECT command shall also implement the MODE SENSE command. MODE SENSE should be issued prior to MODE SELECT to determine supported pages, page lengths, and other parameters.

Table 6-29 Mode Select Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (15h)							
1	Reserved			1 (PF)	Reserved			0 (SP)
2	Reserved							
3	(MSB) Parameter List Length							
4								(LSB)
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

The **Parameter List Length** field specifies the length in bytes of the MODE SELECT parameter list that shall be transferred to the Device as data. A **Parameter List Length** of zero indicates that no data shall be transferred. This condition shall not be considered as an error. A **Parameter List Length** that results in the truncation of any descriptor, header or page of parameters shall cause the Device to terminate the command with “Check Condition” status. The Sense Key shall be set to ILLEGAL REQUEST, and the Additional Sense Code shall be set to PARAMETER LIST LENGTH ERROR.

[Implementor's Note: In some situations where there is a conflict between the information in the command descriptor block and the information in the parameter list, one of several additional sense codes may apply. As a guide, INVALID FIELD IN COMMAND PACKET or PARAMETER LIST LENGTH ERROR should be used if the error is detected prior to any operations that alter the mode parameters. INVALID FIELD IN PARAMETER LIST should be used if the mode parameters have been altered.]

The Device shall terminate the MODE SELECT command with “Check Condition” status, set the Sense Key to ILLEGAL REQUEST and set the Additional Sense Code to INVALID FIELD IN PARAMETER LIST for the following conditions:

1. If an attempt to send an unsupported value in the MODE SELECT header, block descriptor, or any page header.
2. If an attempt to send a page with a length not equal to the parameter length reported for that page by the MODE SENSE command.
3. If an attempt to send a value for a parameter that is outside the range supported by the Device and rounding is not implemented for that parameter.

Table 6-30 Recommended Sense Key, ASC and ASCQ for MODE SELECT Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
05	1A	00	PARAMETER LIST LENGTH ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
05	26	00	INVALID FIELD IN PARAMETER LIST
05	26	01	PARAMETER NOT SUPPORTED
05	26	02	PARAMETER VALUE INVALID
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED

6.8.8. MODE SENSE Command

The MODE SENSE command provides a means for a Device to report parameters. It is a complementary command to the MODE SELECT command.

Table 6-31 Mode Sense Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (1Ah)							
1	Reserved				DBD	Reserved		
2	PC		Page Code					
3	(MSB) Allocation Length							
4	(LSB)							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

A **D**isable **B**lock **D**escriptors (**DBD**) bit of zero indicates that the Device may return zero or more block descriptors in the returned MODE SENSE data, at the Device's discretion. A **DBD** bit of one specifies that the Device shall not return any block descriptors in the returned MODE SENSE data.

The page control (**PC**) field defines the type of parameter values to be returned. The page control field is defined in Table 6-32

Table 6-32 Page Control Field

PC Value	Type of Parameter Values
00b	Current Values
01b	Reserved
10b	Reserved
11b	Reserved

The page code specifies which page or pages to return. Page code usage is defined in Table 6-33.

Table 6-33 Page Code Usage for All Devices

Page Code	Description
00h	Vendor Specific (do not require page format)
01h - 1Fh	See specific Device-types
20h - 3Eh	Vendor Specific (page format required)
3Fh	Return all pages

If a MODE SENSE command is attempted with a page code value not implemented, the Device shall return “Check Condition” status and shall set the Sense Key to ILLEGAL REQUEST and the Additional Sense Code to INVALID FIELD IN COMMAND PACKET.

A page code of 3Fh indicates that all pages implemented by the Device shall be returned. Page 00h, if implemented, shall be returned after all other pages. Devices that implement more than 255 bytes of mode page parameter data and block descriptors shall return “Check Condition” status to a MODE

SENSE request of 3Fh in the page code field. The Sense Key shall be set to ILLEGAL REQUEST and the Additional Sense Code shall be set to INVALID FIELD IN COMMAND PACKET. This limitation ensures compatibility with the SCSI standard.

6.8.8.9. Current Values

A **PC** field value of 0h requests the Device to return the current parameter values for the specified page code. The current values returned are:

1. The parameters set in the last successful MODE SELECT command.
2. The default values if a MODE SELECT command has not been executed since the last power-on, hard RESET condition, or ATAPI Soft Reset.

Table 6-34 Recommended Sense Key, ASC and ASCQ for MODE SENSE Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED

6.8.8.13. Mode Parameters

This section describes the block descriptors and the pages used with MODE SELECT and MODE SENSE commands that are applicable to Devices.

Table 6-35 Mode Page Codes

Page Code	Description	Reference	Type
00h	Vendor specific (do not require page format)		
01h - 0Eh	Reserved		
0Fh	Data Compression Page	6.8.8.55 on page 72	
10h	Reserved		
11h	Medium Partition Page	6.8.8.64 on page 76	Mandatory for QFA
15h - 3Dh	Vendor Specific (page format required)		
2Ah	Capabilities and Mechanical Status	6.8.8.73 on page 77	
3Fh	Returns all pages (valid only for the MODE SENSE command)		

The mode parameter list shown in Table 6-36 contains a header, followed by zero or more block descriptors, followed by zero or more variable-length pages. Parameter lists are defined for each Device type.

Table 6-36 Mode Parameter List

Bit Byte	7	6	5	4	3	2	1	0
0 - n	Mode Parameter Header							
0 - n	Block Descriptor(s)							
0 - n	Page(s)							

Table 6-37 Mode Parameter Header

Bit Byte	7	6	5	4	3	2	1	0
0	Mode Data Length							
1	Medium Type							
2	Device-Specific Parameter							
3	Block Descriptor Length							

When using the MODE SENSE command, the mode data length field specifies the length in bytes of the following data that is available to be transferred. The mode data length does not include itself. When using the MODE SELECT command, this field is reserved.

Implementor's Note: *ATAPI Devices limit the MODE SENSE returned data to 65535 bytes.*

Medium types are unique for each Device type. Refer to the mode parameters section of the specific Device type for definition of these values.

Implementor's Note: *Refer to* Table 6-40 *for a table of medium types for Streaming Tape Devices.*

The Device specific parameter is unique for each Device type. Refer to the mode parameters section of the specific Device type for definition of this field. Some Device types reserve all or part of this field.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include pages or vendor-specific parameters, if

any, that may follow the last block descriptor. A block descriptor length of zero indicates that no block descriptors are included in the mode parameter list. This condition shall not be considered an error.

Table 6-38 Mode Parameter Block Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	Density Code							
1	(MSB)							
2	Number of Blocks							
3								(LSB)
4	Reserved							
5	(MSB)							
6	Block Length							
7								(LSB)

Block descriptors specify some of the medium characteristics. Support for block descriptors is optional. Each block descriptor contains a density code field, a number of blocks field, and a block length field. Block descriptor values are always current (i.e., saving is not supported).

The density code field is unique for each Device type. Refer to the mode parameters section of the specific Device type for definition of this field.

Implementor's Note: *Refer to* Table 6-46 *for a table of density codes for Streaming Tape Devices.*

The number of blocks field specifies the number of logical blocks on the medium to which the density code and block length fields apply. A value of zero indicates that all of the remaining logical blocks of the Device shall have the medium characteristics specified.

Implementor's Note:

(1) There may be implicit association between parameters defined in the pages and block descriptors. In this case, the Device may change parameters not explicitly sent with the MODE SELECT command. A subsequent MODE SENSE command would reflect these changes.

(2) The number of remaining logical blocks may be unknown for some Device types.

The block length specifies the length in bytes of each logical block described by the block descriptor. For QIC Streaming Tape Devices, two block sizes supported may be of either 512 or 1024 bytes per block. Variable length blocks are not supported.

The mode page format is defined in Table 6-39.

Table 6-39 Mode Page Format

Bit Byte	7	6	5	4	3	2	1	0
0	0 (PS)	Reserved	Page Code					
1	Page Length (n - 1)							
2 - n	Mode Parameters							

[**Editor's Note:** Due to hardware restrictions all mode parameters should be padded to a multiple of 4 bytes.]

Each mode page contains a page code, a page length, and a set of mode parameters. The page codes are defined in Table 7-65 and in the mode parameter sub-sections of the specific Device type.

The **Page Code** field identifies the format and parameters defined for that mode page. Some page codes are defined as applying to all Device types and other page codes are defined for the specific Device type.

When using the MODE SENSE command, if page code 00h (vendor-specific page) is implemented, the Device shall return that page last in response to a request to return all pages (page code 3Fh). When using the MODE SELECT command, this page should be sent last.

The **Page Length** field specifies the length in bytes of the mode parameters that follow. If the Host does not set this value to the value that is returned for the page by the MODE SENSE command, the Device shall terminate the command with "Check Condition" status. The Sense Key shall be set to ILLEGAL REQUEST with the Additional Sense Code set to INVALID FIELD IN PARAMETER LIST. The Device is permitted to implement a mode page that is less than the full page length defined by this specification, provided no field is truncated and the page length field correctly specifies the actual length implemented.

The mode parameters for each page are defined in the following sub-sections. Mode parameters not implemented by the Device shall be set to zero.

6.8.8.46.Mode Parameters

This section defines the descriptors and pages for mode parameters used with Devices.

The medium-type code field in the mode parameter header is reserved for use by Devices.

For the MODE SENSE command, the medium-type code field reflects the characteristics of the cartridge currently installed in the Device. For some Devices, the default medium type code value returned in response to a MODE SENSE command may change dynamically to match the most recently inserted medium-type. Table 6-40 defines the medium-type code values

The medium-type code value returned in response to a MODE SENSE command shall be as described below:

1. Following a UNIT ATTENTION condition for a power on or hard reset condition, while not ready, the Device shall report the default or only medium-type.
2. Following a UNIT ATTENTION condition for a not-ready-to-ready transition, the Device shall:
   a) report the default or only medium-type if no attempt has been made by the Device to determine the medium-type.
   b) report the default or only medium-type if the Device cannot automatically determine the medium-type.
   c) report the current recorded medium-type if the Device can automatically determine the medium-type.
3. Following a successful read operation at or after **beginning-of-medium**, the Device shall report a medium-type code value reflecting the installed cartridge. For some implementations, the Device may automatically determine this value. For Devices not capable of automatic medium-type determination, the medium-type code value is provided by the preceding MODE SELECT command.
4. Following an unsuccessful read operation or a successful write operation, while at **beginning-of-partition**, the Device shall:

a) report a medium-type value as described for item (2) if no preceding MODE SELECT command has been issued for the currently loaded cartridge.

b) report a medium-type value as provided by the last successful MODE SELECT command for the currently loaded cartridge.

5. Following a successful unload operation, while not ready, the Device shall report the most recent medium-type code value as determined by items (2) through (4) above.

Table 6-40 ATAPI Streaming Tape Device Medium Type Codes

Code	IPS	Oe	3M Ref No.	Form	Length	A	B	C	D	E	F		
00h	Default												
01h	90	310	DC300	5.25"	300	18	18	36	36	18	18		
02h	90	310	DC300XLP	5.25"	450	18	18	36	36	18	18		
03h	90	550	DC615	5.25"	150	18	18	48	48	18	18		
04h	90	550	DC600	5.25"	620	18	18	48	48	18	18		
05h	120	550	DC6037	5.25"	150	18	18	60	60	18	18		
06h	120	550	DC6150	5.25"	620	18	18	60	60	18	18		
07h	120	550	DC6250	5.25"	1020*	18	18	60	60	18	18		
08h	120	550	DC6320	5.25"	620	18	18	60	60	23	18		
09h	120	550	DC6525	5.25"	1020*	18	18	60	60	23	18		
10h	90	550	DC2000	3.5"	205	12	12	30	30	12	12		
11h	90	550	DC2080	3.5"	205	12	12	30	30	12	12		
12h	120	550	DC2110	3.5"	205	12	15	30	30	12	12		
13h	120	550	DC2120	3.5"	307.5*	12	12	36	36	12	12		
14h	120	550	DC2165	3.5"	307.5*	12	15	36	36	12	12		
15h	120	550	QIC-EST	3.5"	1100	12	12	54	54	12	12		
16h-1Fh	120	900	Reserved	5.25"	TBD								
						A	B	C	G	H	E	F	I
20h-21h	120	900	Reserved	5.25"	TBD								
22h	120	900	DC9210	5.25"	950	18	18	60	60	30	18	18	22o†
23h	120	900	DC9135	5.25"	760	18	18	60	60	30	18	18	23o†
24h	120	900	DC9100	5.25"	760	18	18	60	60	30	18	18	24o†
25h	120	900	DC9120	5.25"	950	18	18	60	60	30	18	18	25o†
26h	120	900	DC9100 SL	5.25"	155	18	18	60	60	30	18	18	26o†
27h	120	900	DC9164	5.25"	950	18	18	60	60	30	18	18	27o†
28h-2Fh			Reserved										
30h	120	900	Firmware	5.25"	NA	18	18	60	60	30	18	18	30o†
31h	120	900	DC9200 SL	5.25"	155	18	18	60	60	30	18	18	31o†
32h	120	900	Reserved	5.25"	TBD								
33h	120	900	DC13GB	5.25"	1200	18	18	60	60	30	18	18	33o†
34h	120	900	DC9200	5.25"	950	18	18	60	60	30	18	18	34o†
35h	120	900	DC9120 XL	5.25"	1200	18	18	60	60	30	18	18	35o†
36h	120	900	DC9210 SL	5.25"	155	18	18	60	60	30	18	18	36o†
37h	120	900	Reserved	5.25"	TBD								
38h-3Fh			Reserved										
40h	120	900	DC9250 XL	5.25"	1200	18	18	60	60	30	18	18	40o†
41h	120	900	Reserved	5.25"	TDB								
42h	120	900	DC9500	5.25"	1200	18	18	60	60	30	18	18	42o†
43h	120	900	Wide Tape	5.25"	1500	18	18	60	60	30	18	18	43o†
44h	120	900	Diagnostic	5.25"	155	18	18	60	60	30	18	18	44o†
45h	120	900	9210, 9200	5.25"	1500	18	18	60	60	30	18	18	45o†
46h	120	900	DC9500 SL	5.25"	155	18	18	60	60	30	18	18	46o†
47h	120	900	DC13GB	5.25"	155	18	18	60	60	30	18	18	47o†
48h-7Fh	120	TBD	Reserved	5.25"	TBD								
						A	B	C	D	E	F		H
80h	120	TBD	Reserved	3.5"	TBD								
81h	120	900	QIC-EST	3.5"	1100	12	27.5	24	24	15	12		01b‡
82h	120	TDB	Reserved	3.5"	TBD								02b‡
83h	120	900	DC2500 LL	3.5"	295	12	27.5	24	24	15	12		03b‡
84h	120	900	DC 3 GB	3.5"	295	12	27.5	24	24	15	12		04b‡

82h	120	TDB	Reserved	3.5"	TBD								02b‡
84h	120	900	DC 3 GB	3.5"	295	12	27.5	24	24	15	12		04b‡
85h	120	TDB	Reserved	3.5"	TBD								05b‡
86h	120	900	Wide Tape	3.5"	400	12	27.5	24	24	15	12		06b‡
87h-90h	120	TBD	Reserved	3.5"	TBD								
91h	120	900	Test	3.5"	100	12	27.5	24	24	15	12		11b‡
92h	120	TBD	BaFe	3.5"	400	24	24	42	42	24	24		12b‡
93h	120	900	DC2500 XL	3.5"	400	12	27.5	24	24	15	12		13b‡
94h-95h	120	TDB	Reserved	3.5"	TBD								
96h	120	900	Wide Tape	3.5"	300	12	27.5	24	24	15	12		16b‡
97h-A0h	120	TDB	Reserved	3.5"	TBD								
A1h	120	900	Firmware	3.5"	100	12	27.5	24	24	15	12		21b‡
A2h-C2h	120	TBD	Reserved	3.5"	TBD								
C3h	120	900		3.5"	300	18	18	72	72	18	18		00b‡
C4h-C5h	120	TBD	Reserved	3.5"	TBD								
C6h	120	900	Wide Tape	3.5"	400	18	18	72	72	18	18		00b‡
C7h-D2h			Reserved		TBD								
D3h	120	900		3.5"	400	18	18	72	72	18	18		00b‡
D4h-FFh			Reserved		TBD								

* Tape Lengths are for extended length cartridges. The hole spacing for standard and extended length cartridges are the same, thus in order to distinguish extended length cartridges from standard length cartridges, a conditioning pass must be performed during which the tape can be measured. If this measurement is not made at cartridge insertion time, the Device shall report the code for the standard length cartridge.

† The hole pattern number determined by the Device is an octal number and must be converted to hex for the medium type. Note that the conversion is aesthetic only and not mathematical. Example 22o = 22h.

‡ The hole pattern number determined by the Device is a 5 bit binary number. To convert it to the medium type value one must add 80h.

Table 6-41 Key for Medium Type Codes

Code	Meaning
IPS	Maximum Speed in Inches per second
Oe	Oersteds
Form	Form Factor in inches
Length	Tape Length in feet.

The codes A through I represent distances (in inches) between holes in the media (center of hole to center of hole). These representations are different for 5.25" and 3.5" cartridges. The meanings of these codes are provided in Table 6-42 and Table 6-43.

NOTE: The terms 1st, 2nd, 3rd, and 4th are counted starting from the **BOT** end of tape.

Table 6-42 Key for 5.25" cartridges:

Code	Meaning
A	2nd and 3rd EOT holes
B	1st and 2nd EOT holes
C	early warning hole to 1st EOT hole
D	load point hole to 3rd BOT hole
E	2nd BOT hole to 3rd BOT hole
F	1st BOT hole to 2nd BOT hole
G	load point hole to 4th BOT hole
H	3rd BOT hole to 4th BOT hole
I	octal hole pattern on cartridge

Table 6-43 Key for 3.5" cartridges:

Code	Meaning
A	1st BOT hole to 2nd BOT hole
B	2nd BOT hole to 3rd BOT hole
C	3rd BOT hole to load point hole
D	early warning to 1st EOT hole
E	1st EOT hole to 2nd EOT hole
F	2nd EOT hole to 3rd EOT hole
G	five bit binary number describing hole pattern

Table 6-44 Device-Specific Parameter

Bit	7	6	5	4	3	2	1	0
	WP	Buffered Mode			Speed			

For the MODE SENSE command, a write protect (**WP**) bit of zero indicates that the medium is write enabled. A **WP** bit of one indicates that the medium is write protected. For the MODE SELECT command, this field is ignored.

The **Buffered Mode** has only one valid value. A value of 1h, the default mode, indicates a write operation will be buffered. All other values are reserved.

Table 6-45 Speed Field Assignments

0h	Default (Use the peripheral Device's default speed).
1h	Use the peripheral Device's lowest speed.
2h - Fh	Use increasing peripheral Device speeds.

Implementor's Note: *It is recommended that the default value for the speed field be 0h. This is the minimum support required. Note that the default speed should consider the Device characteristics, the cartridge specifications, and the tape format.*

For the MODE SELECT command, the density code field indicates the density selected by the Host for use in subsequent read and write operations. For Devices capable of automatic density recognition, the density code selected by the Host may be overridden by the Device for a subsequent read operation if the selected value does not match the current recorded density of the medium.

For the MODE SENSE command, the density code field reflects the current operating density of the Device. For some Devices, the default density code value returned in response to a MODE SENSE command may change dynamically to match the most recently selected density. The density code value returned in response to a MODE SENSE command shall be as described below:

1. Following a UNIT ATTENTION condition for a power on or hard reset condition, while not ready, the Device shall report the default density.
2. Following a UNIT ATTENTION condition for a not-ready-to-ready transition, the Device shall:
   a) report the default density if no attempt has been made by the Device to determine the density.
   b) report the default density if the Device cannot automatically determine the density from the medium.
   c) report the current recorded density if the Device can automatically determine the density from the medium.
3. Following a successful read operation at or after **beginning-of-medium**, the Device shall report a density code value reflecting the recorded density of the medium. For some implementations, the Device may automatically determine this value from the medium. For Devices not capable of automatic density determination, the default density is reported if the density code value is not provided by the preceding MODE SELECT command.
4. Following an unsuccessful read operation or a successful write operation, while at **beginning-of-partition**, the Device shall:
   a) report a density code value as described for item (2) if a previous MODE SELECT command has not established a density code for the currently loaded cartridge.
   b) report a density code value as provided by the last successful MODE SELECT command for the currently loaded cartridge.
5. Following a successful unload operation the Device shall report the most recent density code value as determined by items (2) through (4) above.

Table 6-46 ATAPI Streaming Tape Device Density Codes

Code	mm (Inch)	Tracks	bpmm	bpi	Code	Type	Reference	Note
00h	Default							
01h	12.7 (0.5)	9	32	(800)	NRZI	R	X3.22-1983	2
02h	12.7 (0.5)	9	63	(1600)	PE	R	X3.39-1986	2
03h	12.7 (0.5)	9	246	(6250)	GCR	R	X3.54-1986	2
04h	6.3 (0.25)	4/9	315	(8000)	GCR	C	X3.136-1986	1,3
05h	6.3 (0.25)	4/9	315	(8000)	GCR	C	X3.136-1986	1
06h	12.7 (0.5)	9	126	(3200)	PE	R	X3.157-1987	2,4
07h	6.3 (0.25)	4	252	(6400)	IMFM	C	X3.116-1986	1
08h	3.81 (0.15)	4	315	(8000)	GCR	CS	X3.158-1987	1,4
09h	12.7 (0.5)	18	1491	(37871)	GCR	C	X3B5/87-099	2,4
0Ah	12.7 (0.5)	22	62	(6667)	MFM	C	X3B5/86-199	1,4
0Bh	6.3 (0.25)	4	63	(1600)	PE	C	X3.56-1986	1
0Ch	12.7 (0.5)	24	500	(12690)	GCR	C	HI-TC1	1,6
0Dh	12.7 (0.5)	24	999	(25380)	GCR	C	HI-TC2	1,6
0Eh	Reserved for ECMA							
0Fh	6.3 (0.25)	15	394	(10000)	GCR	C	QIC-120	1,6
10h	6.3 (0.25)	18	394	(10000)	GCR	C	QIC-150	1,6
11h	6.3 (0.25)	26	630	(16000)	GCR	C	QIC-320	1,6
12h	6.3 (0.25)	30	2034	(51667)	RLL	C	QIC-1350	1,6
13h	3.81 (0.15)	1	2400	(61000)	DDS	CS	X3B5/88-185A	5
14h	8.00 (0.315)	1	2126	(54000)		CS	X3B5/88-036	5
15h	6.3 (0.25)	30	1772	(45000)	GCR	C	QIC-1000	1,6
16h - 1Bh	Reserved for 8mm							
1Ch	6.3 (0.25)	40	1654	(42000)	FMFM	C	QIC-3020-MC	1,6
1Dh	6.3 (0.25)	40	1600	(40640)	GCR	C	QIC-3030-MC	1,6
1Eh	6.3 (0.25)	30	630	(67733)	GCR	C	QIC-1000C	1,6
1Fh	6.3 (0.25)	30	2667	(67733)	RLL	C	QIC-2100C	1,6
20h	6.3 (0.25)	144	2667	(67733)	RLL	C	QIC-3070-MC	1,6
21h	6.3 (0.25)	144	2667	(67733)	RLL	C	QIC-5010-DC	1,6
22h	6.3 (0.25)	42	1600	(40640)	GCR	C	QIC-2GB(C)	1,6
23h	6.3 (0.25)	42	1600	(40640)	GCR	C	QIC-3040-MC	1,6
24h - 27h	Reserved							
28h	6.3 (0.25)	44	3264	(82909)	RLL	C	QIC-5GB(C)	1,6
29h	6.3 (0.25)	60	2362	(60000)	RLL	C	QIC-3080-MC	1,6
2Ah - 7Eh	Reserved							
tbd	6.3 (0.25)	20	394	(10000)	FMFM	C	QIC-40	1,6
tbd	6.3 (0.25)	28	579	(14700)	FMFM	C	QIC-80	1,6
tbd	6.3 (0.25)	40	871	(22125)	FMFM	C	QIC-3010	1,6
7Fh	No change from previous density (NO-OP)							7
80h-91h	Vendor unique							
92h	6.3 (0.25)	40	2034	(51667)	RLL	C	QIC-3050-MC	1,6
93h	6.3 (0.25)	48	3675	(93333)	RLL	C	QIC-3110-MC	1,6
80h - FFh	Vendor unique							

Table 6-47 Key for Density Codes

Code	Meaning
NRZI	Non Return to Zero, change on ones
GCR	Group Code Recording
PE	Phase Encoded
MFM	Modified Frequency Modulation
IMFM	Inverted MFM
FMFM	Floppy compatible MFM
DDS	DAT Data Storage
RLL	Run Length Limited
R	Reel-to-Reel
C	Cartridge
CS	Cassette

NOTES:

1. Serial Recorded.
2. Parallel Recorded.
3. Old format known as QIC-11.
4. See Appendix D for additional standards information.
5. Helical Scan
6. This is not an American National Standard. The reference is based on an industry standard definition of the media format.
7. This density code value is defined for the MODE SELECT command and shall not be returned by the MODE SENSE command.

6.8.8.55.Data Compression Page Code 0Fh

Table 6-48 Data Compression Page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page Code (0Fh)					
1	Page Length (0Eh)							
2	DCE	DCC	Reserved					
3	DDE	RED		Reserved				
4	(MSB)							
5	Compression Algorithm							
6								
7								(LSB)
8	(MSB)							
9	Decompression Algorithm							
10								
11								(LSB)
12	Reserved							
13	Reserved							
14	Reserved							
15	Reserved							

This page specifies the parameters for the control of data compression in a Device.

A **D**ata **C**ompression **E**nable (**DCE**) bit of one indicates that the data compression is to be enabled. When this bit is set, data sent to the Device by the Host shall be processed using the selected compression algorithm before being written to the medium. A **DCE** bit of zero indicates that data compression is disabled.

A **D**ata **C**ompression **C**apable (**DCC**) bit of one indicates that the Device supports data compression and shall process data sent to it for transferal to the medium using the selected compression algorithm when the **DCE** bit is one. A **DCC** bit of zero indicates that the Device does not support data compression. This shall be a non-changeable field.

A **D**ata **D**ecompression **E**nable (**DDE**) bit of one indicates that data decompression is to be enabled. A **DDE** of zero indicates that data decompression is disabled.

Implementor's Note: *When the **DDE** bit is zero, all decompression algorithms are deemed unsupported by the Device.*

The **R**eport **E**xception on **D**ecompression (**RED**) field indicates the Device's response to certain boundaries it detects in the data on the medium. There are a number of boundaries that may occur on the medium between compressed and uncompressed data. These boundaries are shown in Table 6-49.

Table 6-49 Possible data compression boundaries

Prior Data	Current Data
uncompressed	compressed (unsupported algorithm)
uncompressed	compressed (supported algorithm)
compressed (supported algorithm)	uncompressed
compressed (supported algorithm)	compressed (unsupported algorithm)
compressed (supported algorithm A)	compressed (supported algorithm B)
compressed (unsupported algorithm)	uncompressed
compressed (unsupported algorithm)	compressed (supported algorithm)
compressed (unsupported algorithm A)	compressed (unsupported algorithm B)

A **RED** field of zero indicates that the Device shall return a “Check Condition” status when data is encountered on the medium during a read operation that the Device cannot decompress. This is the case at the boundaries shown in Table 6-50.

Table 6-50 Boundaries that generate a “Check Condition” when RED field is zero

Prior Data	Current Data	Sense Key
uncompressed	compressed (unsupported algorithm)	MEDIUM ERROR
compressed (supported algorithm)	compressed (unsupported algorithm)	MEDIUM ERROR
compressed (unsupported algorithm A)	compressed (unsupported algorithm B)	MEDIUM ERROR

A **RED** field of one indicates that the Device shall return a “Check Condition” status when data in encountered on the medium during a read operation that requires different handling by the Host than the

data most recently encountered during a prior read operation. This is the case in the boundaries shown in Table 6-51.

Table 6-51 Boundaries that generate a "Check Condition" when RED field is one

Prior Data	Current Data	Sense Key
uncompressed	compressed (unsupported algorithm)	MEDIUM ERROR
compressed (supported algorithm)	compressed (unsupported algorithm)	MEDIUM ERROR
compressed (unsupported algorithm)	uncompressed	NO SENSE
compressed (unsupported algorithm)	compressed (supported algorithm)	RECOVERED ERROR
compressed (unsupported algorithm A)	compressed (unsupported algorithm B)	MEDIUM ERROR

At each of these boundaries, the data sent to the Host is of a fundamentally different nature from that which was previously sent.

A **RED** field of two indicates that the Device shall return a "Check Condition" status when data is encountered on the medium during a read operation that has been processed using a different algorithm from that data most recently encountered during a prior read operation. This is the case at the boundaries shown in Table 6-52.

Table 6-52 Boundaries that generate a "Check Condition" when RED field is zero

Prior Data	Current Data	Sense Key
uncompressed	compressed (unsupported algorithm)	MEDIUM ERROR
uncompressed	compressed (supported algorithm)	RECOVERED ERROR
compressed (supported algorithm)	uncompressed	NO SENSE
compressed (supported algorithm)	compressed (unsupported algorithm)	MEDIUM ERROR
compressed (supported algorithm A)	compressed (supported algorithm B)	RECOVERED ERROR
compressed (unsupported algorithm)	uncompressed	NO SENSE
compressed (unsupported algorithm)	compressed (supported algorithm)	RECOVERED ERROR
compressed (unsupported algorithm A)	compressed (unsupported algorithm B)	MEDIUM ERROR

On any of the boundary conditions that results in a "Check Condition" status, the Additional Sense Code shall be set to either DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN (id the algorithm identifier & 255) or DECOMPRESSION EXCEPTION LONG ALGORITHM ID. The Device shall, in both cases, set the decompression algorithm field to the algorithm identifier of the compression algorithm used to process the encountered data. The Device shall be positioned on the **EOP** side of the encountered data, and the Command-Specific Information field in the sense data shall contain the count of the number of data blocks contained within the encountered data.

Implementor's Note: *When compressed data is encountered on the medium that the Device cannot decompress, the Device should treat the data as uncompressed. In the sense data, the Valid bit, the ILI bit, and the Information field should be set accordingly.*

A **RED** field of 3 is undefined and shall result in a "Check Condition" status with the Sense Key set to ILLEGAL REQUEST.

The **Compression algorithm** field indicates the compression algorithm the Device shall use to process the data sent to it by the Host when the DCE bit is set to one. If the Host selects an algorithm the Device does not support the Device shall return a "Check Condition" status. The Sense Key shall be set to ILLEGAL REQUEST. A value of zero shall indicate that no compression algorithm is currently selected.

For the MODE SELECT command, the **Decompression Algorithm** field indicates the decompression algorithm selected by the Host for use in subsequent decompression of data encountered on the medium. For Devices capable of the automatic recognition of the compression algorithm used to process data encountered on the medium, the decompression algorithm selected by the Host may be ignored, or overridden by the Device for a subsequent read operation if the selected value does not match the compression algorithm detected by the Device, which was used to process the data encountered on the medium.

For the MODE SENSE command, the **Decompression Algorithm** field reflects the algorithm selected by the Host. For some Devices, the **Decompression Algorithm** value returned in response to a MODE SENSE command may change dynamically to match the compression algorithm, detected by the Device, which was used to process the data most recently encountered on the medium, during a read operation. A value of zero shall indicate the data encountered on the medium during the most recent read operation was uncompressed. Table 6-53 shows the compression algorithm identifiers registered under the International Register of Processing Algorithms [to be] established by ISO/IEC JTC1.

Table 6-53 Compression Algorithm Identifier

Algorithm Identifier	Description
00h	No Algorithm Selected (Identifies uncompressed data)
01h	Unused
02h - 0Fh	Not assigned
10h	IBM IDRC Data Compaction Algorithm
11h - 1Fh	Not assigned
20h	DCLZ Data Compaction Algorithm
21h - FDh	Not assigned
FEh	Reserved
FFh	Unregistered Algorithm
100h - FFFFFFFFh	Reserved

6.8.8.64.Medium Partition Page Code 11h

Table 6-54 Medium Partition Page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page Code (11h)					
1	Page Length (6)							
2	0 (Maximum Additional Partitions)							
3	0 (Additional Partitions Defined)							
4	FDP	0 (SDP)	0 (IDP)	00 (PSUM)		Reserved		
5	Medium Format Recognition							
6	Reserved							
7	Reserved							

This page is used to specify the medium partitions.

A **F**ixed **D**ata **P**artitions (**FDP**) bit of one indicates that the Device assigns partitions based on its fixed definition of partitions. Setting this bit to one may only be valid at **beginning-of-partition**.

Implementor's Note:

1. *The **FDP** bit is used to indicate that the Device will assign partitions based on a pre-defined (format-specific) definition of the data and directory partitions.*
2. *If the **FDP** bit is set to one, the Device shall implement a maximum of two partitions for the QFA function (a data partition 0 and a directory partition 1).*
3. *The **FDP** bit cannot be changed when the tape is positioned away from **BOT**.*
4. *Devices indicate support for QFA through the Capabilities and Mechanical Status page.*
5. *If the Device can detect that an inserted cartridge has been previously recorded as a QFA tape, the Device shall report the **FDP** bit as a one in Mode Sense. If the Host issues a write command from the BOT position without first issuing a Mode Select or Mode Sense, the **FDP** bit is reported as a zero and the tape will not be partitioned. This may require new bursts be written.*

***Implementor's Note:** Since defining partitions may require re-formatting the medium for some implementations, an implicit write to the medium may occur as a result of a MODE SELECT command that supplies any of these parameters.*

The **Medium Format Recognition** field is a Device-defined value indicating the Device's capability to automatically identify the medium format and partition information when reading an unknown volume.

Table 6-55 Medium Format Recognition Field

Value	Meaning
00h	Incapable of format or partition recognition.
01h	Capable of format recognition only.
02h	Capable of partition recognition only.
03h	Capable of format and partition recognition.
04h-FFh	Reserved.

Implementor's Note: *If a Device indicates that it is not capable of medium format recognition, the Host must supply all necessary parameters for the Device to identify the specific format. The value in this field may be different following a medium change.*

6.8.8.73.Capabilities and Mechanical Status Page Code 2Ah

Table 6-56 Capabilities and Mechanical Status Page

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved		Page Code (2Ah)					
1	Page Length (12h)							
2	Reserved							
3	Reserved							
4	Reserved	Reserved	SPREV	Reserved	Reserved	Reserved	Reserved	RO
5	Reserved	Reserved	QFA	Reserved	EFMT	Reserved	Reserved	Reserved
6	CMPRS	ECC	Reserved	Reserved	EJECT	PREVENT	LOCKED	LOCK
7	SLOWB	Reserved	Reserved	Reserved	Reserved	BLK1024	BLK512	Reserved
8	(MSB) Maximum Speed Supported (in KBps)							
9	(LSB)							
10	(MSB) Maximum Stored Defected List Entries							
11	(LSB)							
12	(MSB) Continuous Transfer Limit (in blocks)							
13	(LSB)							
14	(MSB) Current Speed Selected (in Kbps)							
15	(LSB)							
16	(MSB) Buffer Size (in 512 bytes)							
17	(LSB)							
18-19	Reserved							

If the **SPREV** bit is set, the Devices supports SPACE in the reverse direction.

If the **RO** bit is set, the Device is operating in a read only mode.

If the **QFA** bit is set, the Device supports the QFA two partition formats.

If the **EFMT** bit is set, the Device supports ERASE command initiated formatting.

If the **CMPRS** bit is set, the Device supports data compression.

If the **ECC** bit is set, the Device supports error correction.

If the **EJECT** bit is set, the Device can mechanically unload the volume with the LOAD/UNLOAD command.

If the **PREVENT** bit is set, the Device defaults in the Prevent state after power up.

If the **LOCKED** bit is set, the volume is locked.

If the **LOCK** bit is set, the Device supports locking the volume using the ALLOW/PREVENT MEDIUM REMOVAL command.

If the **SLOWB** bit is set, the Device restricts the byte count for PIO transfers for slow buffer memory.

If the **BLK1024** bit is set, the Device supports 1024 byte block size.

If the **BLK512** bit is set, the Device supports 512 byte block size.

The **Maximum Speed Supported** field indicates the actual maximum data that the Device supports. This value is returned as 1000 bytes per second that the data is transferred between the Host and the Device.

The **Continuous Transfer Limit** field indicates the number of blocks for the current block size that can be transferred without delay due to a buffer limitation. The DSC bit in the Status Register set indicates that the specified number of blocks can be transferred without delay. This information allows techniques that maximize the bus bandwidth.

The **Current Speed Selected** field indicates the actual data rate that the Device is currently using this value is return as 1000 bytes per second that the data is transferred between the Host and the Device.

The **Buffer Size** is an estimate in 512 byte increments of the read and write buffer size to aid in establishing meaningful buffer ratios in the Device Configuration Page.

6.8.9. READ Command

The READ command requests that the Device transfer one or more block(s) of data to the Host beginning with the current block.

Table 6-57 Read Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (08h)							
1	Reserved						Reserved (SILI)	1 (Fixed)
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

The **Transfer Length** specifies the number of fixed-length blocks to be transferred, using the current block length reported in the mode parameters block descriptor.

A successful READ command shall transfer the requested **Transfer Length** times the current block length in bytes. Upon completion, the logical position shall be after the last block transferred (**end-of-partition** side).

If an incorrect length block is read, “Check Condition” status shall be returned and the ILI and Valid bits shall be set to one in the sense data. Upon termination, the logical position shall be after the incorrect length block (**end-of-partition** side). The Information field shall be set to the difference (residue) of the requested **Transfer Length** minus the actual number of blocks read (not including the incorrect length block).

A **Transfer Length** of zero indicates that no data shall be transferred. This condition shall not be considered an error and the logical position shall not be changed but the Device shall start a read ahead mode of operation. The DSC bit is set when sufficient data is available in the ATAPI device’s data buffer.

If the Device encounters a filemark during a READ command, “Check Condition” status shall be returned and the filemark and Valid bits shall be set to one in the sense data. The Sense Key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. Upon termination, the logical position shall be after the filemark (**end-of-partition** side). The Information field shall be set to the difference (residue) of the requested **Transfer Length** minus the actual number of blocks read (not including the filemark).

Upon termination, the logical position shall be after the last block transferred or the block with an unrecovered read error (**end-of-partition** side). The Information field shall be set to the difference (residue) of the requested **Transfer Length** minus the actual number of blocks read.

If the Device encounters **end-of-data** during a READ command, “Check Condition” status shall be returned, the Sense Key shall be set to BLANK CHECK, and the Valid bit shall be set to one in the

sense data. If **end-of-data** is encountered at or after **early-warning**, the EOM bit shall also be set to one. Upon termination, the logical position shall be after the last recorded logical block (**end-of-partition** side). The Information field shall be set to the difference (residue) of the requested **Transfer Length** minus the actual number of blocks read.

If the Device encounters **end-of-partition** during a READ command, "Check Condition" status shall be returned, the Sense Key shall be set to MEDIUM ERROR, and the EOM and Valid bits shall be set to one in the sense data. The medium position following this condition is not defined. The Information field shall be set to the difference (residue) of the requested **Transfer Length** minus the actual number of blocks read.

Table 6-58 Recommended Sense Key, ASC and ASCQ for READ Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
03	11	00	UNRECOVERED READ ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.10. READ POSITION Command

The READ POSITION command reports the current logical position of the Device. No medium movement shall occur as a result of the command.

Table 6-59 Read Position Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (34h)							
1	Reserved							0 (BT)
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

A **B**lock **T**ype (**BT**) bit of zero indicates the value(s) returned in the Read Position Data , First and Last Block Location fields shall be interpreted as a sequential logical block address(es) with block zero originating at the **beginning-of-partition**.

If the Device is not capable of reporting block locations, the **BPU** bit shall be set to one in the returned data. A REWIND/LOCATE operation may recover from this condition.

The READ POSITION data shown in Table 6-60 shall be returned as data after the command.

Table 6-60 Read Position Data Format

Bit Byte	7	6	5	4	3	2	1	0
0	BOP	EOP	Reserved			BPU	Reserved	
1	Partition Number							
2	Reserved							
3	Reserved							
4	(MSB)							
5	First Block Location							
6								
7								(LSB)
8	(MSB)							
9	Last Block Location (Optional)							
10								
11								(LSB)
12	Reserved							
13	(MSB)							
14	Blocks in Buffer (Optional)							
15								(LSB)
16	(MSB)							
17	Bytes in Buffer (Optional)							
18								
19								(LSB)

A **B**eginning **O**f **P**artition (**BOP**) bit of one indicates that the Device is at the **beginning-of-partition** in the current partition. A **BOP** bit of zero indicates that the current logical position is not at the **beginning-of-partition**.

An **E**nd **O**f **P**artition (**EOP**) bit of one indicates that the Device is positioned between **early-warning** and **end-of-partition** in the current partition. An **EOP** bit of zero indicates that the current logical position is not between **early-warning** and **end-of-partition**.

***Implementor's Note**: The **BOP** and **EOP** indications are not necessarily a result of a physical tape marker (e.g., reflective marker or hole).*

A **B**lock **P**osition **U**nknown (**BPU**) bit of one indicates that the block location is not known or cannot be obtained. A **BPU** bit of zero indicates that the block location field contains valid position information.

The **Partition Number** field reports the partition number for the current logical position. If the Device only supports one partition for the medium, this field shall be set to zero.

The **First Block Location** field indicates the block address associated with the current position. This value shall indicate the block address of the data block to be transferred between the Host and the Device if a READ or WRITE command is issued.

The **Last Block Location** field indicates the block address associated with the next block to be transferred from the buffer to the medium. The value shall indicate the block address of the next data block to be transferred between the buffer and the medium. If the buffer does not contain a whole block of data or is empty, the value reported for the **Last Block Location** shall be equal to the value reported for the **First Block Location**.

The **Blocks in Buffer** field indicate the number of blocks that are in the Device buffer. Following a

write operation, this field indicates the block number of blocks not written to the medium. Whereas, as an ATAPI extension to the SCSI standard, during a read operation this field indicates the number of blocks read ahead in the buffer.

The **Bytes in Buffer** field indicate the number of bytes in the Device buffer. This information can be used to determine the number of filemarks counted as blocks in the **Blocks in Buffer** field and shall always represent whole blocks of data.

Table 6-61 Recommended Sense Key, ASC and ASCQ for READ POSITION Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
04	15	00	RANDOM POSITIONING ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.11. REQUEST SENSE Command

The REQUEST SENSE command requests that the Device transfer sense data to the Host.

Table 6-62 Request Sense Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (03h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

The sense data:

1. shall be available if an error condition (“Check Condition”) had previously been reported to the Host;
2. shall be available if other information (e.g. medium position) is available in any field.

If the Device has no other sense data available to return, it shall return a Sense Key of NO SENSE and an Additional Sense Code of NO ADDITIONAL SENSE INFORMATION.

The sense data shall be preserved by the Device until retrieved by a REQUEST SENSE command or until the receipt of any other I/O Command.

The Device shall return “Check Condition” status for a REQUEST SENSE command only to report exception conditions specific to the command itself. For example:

1. A non-zero reserved bit is detected in the Command Packet;
2. A Device malfunction prevents return of the sense data.

If a recovered error occurs during the execution of the REQUEST SENSE command, the Device shall return the sense data with “Good” status. If a Device returns “Check Condition” status for a REQUEST SENSE command, the sense data may be invalid.

Devices shall be capable of returning at least 18 bytes of data in response to a REQUEST SENSE command. If the allocation length is 18 or greater, and a Device returns less than 18 bytes of data, the Host should assume that the bytes not transferred would have been zeros had the Device returned those bytes. Host can determine how much sense data has been returned by examining the allocation length parameter in the Command Packet and the additional sense length in the sense data. Devices shall not adjust the additional sense length to reflect truncation if the allocation length is less than the sense data available.

The sense data format for error codes 70h (current errors) and 71h (deferred errors) are defined in Table 6-63 Request Sense Standard Data. Error code values of 72h to 7Eh are reserved. Error code 7Fh is for a vendor-specific sense data format. Devices shall implement error code 70h;

implementation of error code 71h is optional. Error code values of 00h to 6Fh are not defined by this Specification and their use is not recommended.

Table 6-63 Request Sense Standard Data

Bit Byte	7	6	5	4	3	2	1	0
0	Valid	Error Code (70h or 71h)						
1	Reserved (Segment Number)							
2	Filemark	EOM	ILI	Reserved	Sense Key			
3	(MSB)							
4				Information				
5								
6								(LSB)
7	Additional Sense Length (n - 7)							
8-11	Command Specific Information (Optional)							
12	Additional Sense Code							
13	Additional Sense Code Qualifier (Optional)							
14	Field Replaceable Unit Code (Optional)							
15	SKSV (Optional)	Sense Key Specific (Optional)						
16-17	Sense Key Specific (Optional)							
18	Additional Sense Bytes							
n								

[**Editor's Note:** Due to hardware restrictions all sense data should be padded to a multiple of 4 bytes.]

A **Valid** bit of zero indicates that the **Information** field is not as defined in this Specification. A **Valid** bit of one indicates the **Information** field contains valid information as defined in this Specification. Devices shall implement the **Valid** bit.

The Segment Number field is Reserved.

The **Filemark** bit is mandatory for Streaming Tape Devices and this bit is reserved for all other Devices. A **Filemark** bit of one indicates that the current command has read a filemark.

The **end-of-medium** (**EOM**) bit is mandatory for Streaming Tape Devices and printer Devices and this bit is reserved for all other Device types. An **EOM** bit of one indicates that **end-of-medium** condition (**end-of-partition**, **beginning-of-partition**, out-of-paper, etc.) exists. For Streaming Tape Devices, this bit indicates that the unit is at or past the **early-warning** if the direction was forward or that the command could not be completed because **beginning-of-partition** was encountered if the direction was reverse.

An **I**ncorrect **L**ength **I**ndicator (**ILI**) bit of one indicates that the requested allocation length or selected blocks size did not match the logical block length of the data on the medium.

The **Sense Key**, **Additional Sense Code** and **Additional Sense Code Qualifier** provide a hierarchy of information. The intention of the hierarchy is to provide a top-down approach for the Host to determine information relating to the error and exception conditions. The **Sense Key** provides generic categories in which error and exception conditions can be reported. The Host would typically use the Sense Key for high-level error recovery procedures. **Additional Sense Code** provides further detail describing the **Sense Key**. **Additional Sense Code Qualifier** adds further detail to the **Additional Sense Code**. The **Additional Sense Code** and **Additional Sense Code Qualifier** can be used by the Host where sophisticated error recovery procedures require detailed information describing the error and exception conditions.

The **Sense Key** field is mandatory and indicates generic information describing an error or exception condition. The sense keys are defined in Table 6-66 Sense Key Descriptions on page 89.

The content of the **Information** field is command-specific and is defined within the appropriate section for the command of interest. Devices shall implement the **Information** field. Unless specified otherwise, when the **Valid** bit is one, this field contains the difference between the number of blocks requested by the command or commands and the actual number of blocks and filemarks transferred to or from the medium, the residue.

The **Additional Sense Length** field indicates the number of additional sense bytes to follow. If the allocation length of the Command Packet is too small to transfer all of the additional sense bytes, the **Additional Sense Length** is not adjusted to reflect the truncation.

The **Command-Specific Information** field contains information that depends on the command that was executed. Further meaning for this field is defined within the command description.

The **Additional Sense Code** (**ASC**) field indicates further information related to the error or exception condition reported in the **Sense Key** field. Devices shall support the **Additional Sense Code** field. Support of the additional sense codes not explicitly required by this specification is optional. A list of additional sense codes is defined in Table 6-67. If the Device does not have further information related to the error or exception condition, the **Additional Sense Code** is set to NO ADDITIONAL SENSE INFORMATION.

The **Additional Sense Code Qualifier** (**ASCQ**) indicates detailed information related to the **Additional Sense Code**. The **ASCQ** is optional. If the error or exception condition is reportable by the Device, the value returned shall be as specified in the command descriptions. If the Device does not have detailed information related to the error or exception condition, the **ASCQ** is set to zero.

Non-zero values in the **Field Replaceable Unit Code** field are used to define a Device-specific mechanism or unit that has failed. A value of zero in this field shall indicate that no specific mechanism or unit has been identified to have failed or that the data is not available. The **F**ield **Replaceable Unit Code** field is optional. The format of this information is not specified by this Specification.

The **Additional Sense Bytes** field may contain command specific data, peripheral Device specific data, or vendor-specific data that further defines the nature of the “Check Condition” status.

6.8.11.9. Sense-key Specific

The **S**ense-**K**ey **S**pecific field is defined by this Specification when the value of the **S**ense-**K**ey **S**pecific **V**alid (**SKSV**) bit is one. The **S**ense-**K**ey **S**pecific **V**alid bit and **S**ense-**K**ey **S**pecific field are optional. The definition of this field is determined by the value of the **Sense Key** field. This field is reserved for sense keys not described below. An **SKSV** value of zero indicates that this field is not as defined by this Specification.

If the **Sense Key** field is set to ILLEGAL REQUEST and the **SKSV** bit is set to one, the sense-key specific field indicates which illegal parameters in the Command Packet or the data parameters are in error.

Table 6-64 Field Pointer Bytes

Bit Byte	7	6	5	4	3	2	1	0
15	SKSV	C/D	Reserved	Reserved	BPV	Bit Pointer		
16	(MSB)	Field Pointer						
17								(LSB)

A **C**ommand **D**ata (**C/D**) bit of one indicates that the illegal parameter is in the Command Packet. A **C/D** bit of zero indicates that the illegal parameter is in the data parameters sent by the Host.

A **B**it **P**ointer **V**alid (**BPV**) bit of zero indicates that the value in the **Bit Pointer** field is not valid. A **BPV** bit of one indicates that the **Bit Pointer** field specifies which bit of the byte designated by the **Field Pointer** field is in error. When a multiple-bit field is in error, the **Bit Pointer** field shall point to the most-significant (left-most) bit of the field.

The **Field Pointer** field indicates which byte of the Command Packet or of the parameter data was in error. Bytes are numbered starting from zero, as shown in the tables describing the commands and parameters. When a multiple-byte field is in error, the pointer shall point to the most significant (left-most) byte of the field.

If the Sense Key is RECOVERED ERROR, HARDWARE ERROR or MEDIUM ERROR and if the **SKSV** bit is one, the sense-key specific field shall be as shown in Table 6-65 Field Pointer Bytes.

Table 6-65 Field Pointer Bytes

Bit Byte	7	6	5	4	3	2	1	0
15	SKSV	C/D	Reserved	Reserved	BPV	Bit Pointer		
16	(MSB)	Actual Retry Count						
17								(LSB)

The **Actual Retry Count** field returns implementation-specific information on the actual number of retries of the recovery algorithm used in attempting to recover an error or exception condition. This field should relate to the retry count fields within the Error Recovery Page of the MODE SELECT command.

6.8.11.26. Deferred Errors

Error code 70h indicates that the “Check Condition” status returned is the result of an error or exception condition on the I/O process that returned the “Check Condition” status. This includes errors generated during execution of the command by the actual execution process. It also includes errors not related to any command that are first observed during execution of a command. Examples of this latter type of error include a servo-mechanism, off-track errors, and power-up test errors.

Error code 71h (deferred error) indicates that the “Check Condition” status returned is the result of an error or exception condition that occurred during execution of a previous command for which “Good” status has already been returned. Such commands are associated with use of the Immediate bit, with some forms of caching, and with multiple command buffering. Devices that implement these features are required to implement deferred error reporting. For ATAPI Streaming Tape Devices, a Media Access command shall provide an immediate Completion Status with the DSC bit

cleared in the Status Register to free the bus. The actual Completion Status is unknown until the DSC bit in the Register Status is set. This delayed status is not a deferred error.

The deferred error may be indicated by returning “Check Condition” status to the Host as described below. The subsequent execution of a REQUEST SENSE command shall return the deferred error sense information. If an I/O Command terminates with “Check Condition” status and the subsequent sense data returns a deferred error, that I/O command shall **not** have been executed. After the Device detects a deferred error condition, it shall return a deferred error according to the rules described below:

1. If a deferred error can be recovered with no external system intervention, a deferred error indication shall not be set unless required by the error handling parameters of the MODE SELECT command. The occurrence of the error may be logged if statistical or error logging is supported.
2. If a deferred error can be associated with a particular function or a particular subset of data, and the error is either unrecovered or required to be reported by the mode parameters, a deferred error indication shall be returned to the Host.

Deferred errors may indicate that an operation was unsuccessful long after the command performing the data transfer returned “Good” status. If data that cannot be replicated or recovered from other sources is being stored using buffered write operations, synchronization commands should be performed before the critical data is destroyed in the Host. This is necessary to be sure that recovery actions can be taken if deferred errors do occur in the storing of the data.

6.8.11.27. Sense-key and Sense Code Definitions

Table 6-66 Sense Key Descriptions

Sense key	Description
0h	NO SENSE. Indicates that there is no specific Sense Key information to be reported. This would be the case for a successful command.
1h	RECOVERED ERROR. Indicates that the last command completed successfully with some recovery action performed by the Device. Details may be determinable by examining the additional sense bytes and the Information field. When multiple recovered errors occur during one command, the choice of which error to report (first, last, most severe, etc.) is Device specific.
2h	NOT READY. Indicates that the Device cannot be accessed. Operator intervention may be required to correct this condition.
3h	MEDIUM ERROR. Indicates that the command terminated with a non-recovered error condition that was probably caused by a flaw in the medium or an error in the recorded data. This Sense Key may also be returned if the Device is unable to distinguish between a flaw in the medium and a specific hardware failure (Sense Key 4h).
4h	HARDWARE ERROR. Indicates that the Device detected a non-recoverable hardware failure (for example, controller failure, Device failure, parity error, etc.) while performing the command or during a self test.
5h	ILLEGAL REQUEST. Indicates that there was an illegal parameter in the Command Packet or in the additional parameters supplied as data for some commands. If the Device detects an invalid parameter in the Command Packet, then it shall terminate the command without altering the medium. If the Device detects an invalid parameter in the additional parameters supplied as data, then the Device may have already altered the medium.
6h	UNIT ATTENTION. Indicates that the removable medium may have been changed or the Device has been reset.
7h	DATA PROTECT. Indicates that a command that reads or writes the medium was attempted on a block that is protected from this operation. The read or write operation is not performed.
8h	BLANK CHECK. Indicates that while reading either blank medium or format defined **end-of-data** was encountered or, for a write-once device while writing, non-blank medium was encountered.
9h - Ah	Reserved
Bh	ABORTED COMMAND. Indicates that the Device has aborted the command. The Host may be able to recover by trying the command again. This error is reported for conditions such as an overrun etc.
Dh	VOLUME OVERFLOW. Indicates a write operation was unable to complete a transfer.
Eh - Fh	Reserved

6.8.11.29. Using the REQUEST SENSE Command

Whenever an Error is reported, the Host should issue a REQUEST SENSE command to receive the sense data describing what caused the Error condition. If the Host issues some other command, the sense data is lost.

Table 6-67 ASC and ASCQ Assignments

ASC	ASCQ	Description
00	00	NO ADDITIONAL SENSE INFORMATION
00	01	FILEMARK DETECTED
00	02	END-OF-PARTITION/MEDIUM DETECTED
00	04	BEGINNING-OF-PARTITION/MEDIUM DETECTED
00	05	END-OF-DATA DETECTED
03	00	PERIPHERAL DEVICE WRITE FAULT
03	01	NO WRITE CURRENT
03	02	EXCESSIVE WRITE ERRORS
04	00	NOT READY, CAUSE NOT REPORTABLE
04	01	IN PROCESS OF BECOMING READY
04	02	NOT READY, INITIALIZING COMMAND REQUIRED
04	03	NOT READY, MANUAL INTERVENTION REQUIRED
04	04	NOT READY, FORMAT IN PROGRESS
08	00	COMMUNICATION FAILURE
08	01	COMMUNICATION TIME-OUT
09	00	TRACK FOLLOWING ERROR
0A	00	ERROR LOG OVERFLOW
0C	00	WRITE ERROR sense key says whether recovered
11	00	UNRECOVERED READ ERROR
11	01	READ RETRIES EXHAUSTED
11	02	ERROR TOO LONG TO CORRECT
11	03	MULTIPLE READ ERRORS
11	08	INCOMPLETE BLOCK READ (POSTAMBLE NOT FOUND)
11	09	NO GAP FOUND
11	0A	MISCORRECTED ERROR
14	00	RECORDED ENTITY NOT FOUND
14	01	RECORD NOT FOUND
14	02	FILEMARK NOT FOUND
14	03	END-OF-DATA NOT FOUND
14	04	BLOCK SEQUENCE ERROR
15	00	RANDOM POSITIONING ERROR
15	01	MECHANICAL POSITIONING ERROR
15	02	POSITIONING ERROR DETECTED BY READ OF MEDIUM
17	00	RECOVERED DATA WITH NO ERROR CORRECTION APPLIED
17	01	RECOVERED DATA WITH RETRIES
17	02	RECOVERED DATA WITH POSITIVE HEAD OFFSET
17	03	RECOVERED DATA WITH NEGATIVE HEAD OFFSET
18	00	RECOVERED DATA WITH ERROR CORRECTION APPLIED
19	00	(DEFECT LIST ERROR*)
1A	00	PARAMETER LIST LENGTH ERROR
20	00	INVALID COMMAND OPERATION CODE
21	00	LOGICAL BLOCK ADDRESS OUT OF RANGE
24	00	INVALID FIELD IN COMMAND PACKET check field pointer in sense data
25	00	LOGICAL UNIT NOT SUPPORTED
26	00	INVALID FIELD IN PARAMETER LIST check field pointer is sense data
26	01	PARAMETER NOT SUPPORTED check field pointer is sense data
26	02	PARAMETER VALUE INVALID check field pointer in sense data
26	03	THRESHOLD PARAMETERS NOT SUPPORTED
27	00	WRITE PROTECTED
28	00	NOT READY TO READY TRANSITION medium may have changed

2A	00	PARAMETERS CHANGED
2A	01	MODE PARAMETERS CHANGED
2D	00	OVERWRITE ERROR ON UPDATE IN PLACE
29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
30	00	INCOMPATIBLE MEDIUM INSTALLED
30	01	CANNOT READ MEDIUM - UNKNOWN FORMAT
30	02	CANNOT READ MEDIUM - INCOMPATIBLE FORMAT
30	03	CLEANING CARTRIDGE INSTALLED
31	00	MEDIUM FORMAT CORRUPTED
33	00	TAPE LENGTH ERROR
37	00	ROUNDED PARAMETER
39	00	SAVING PARAMETERS NOT SUPPORTED
3A	00	MEDIUM NOT PRESENT
3B	00	SEQUENTIAL POSITIONING ERROR
3B	01	TAPE POSITION ERROR AT BEGINNING-OF-MEDIUM
3B	02	TAPE POSITION ERROR AT END-OF-MEDIUM
3B	08	REPOSITION ERROR
3E	00	NOT SELF-CONFIGURED YET
3F	00	OPERATING CONDITIONS HAVE CHANGED
3F	01	MICROCODE HAS BEEN CHANGED
3F	03	INQUIRY DATA HAS CHANGED
40	NN	DIAGNOSTIC FAILURE ON COMPONENT NN (80H-FFH)
44	00	INTERNAL FAILURE
46	00	UNSUCCESSFUL SOFT RESET
4C	00	FAILED SELF-CONFIGURATION
4E	00	OVERLAPPED COMMANDS ATTEMPTED
50	00	WRITE APPEND ERROR
50	01	WRITE APPEND POSITION ERROR
50	02	POSITION ERROR RELATED TO TIMING
51	00	ERASE FAILURE (FORMAT FAILURE*)
52	00	CARTRIDGE FAULT
53	00	MEDIA LOAD/EJECT FAILED
53	01	UNLOAD TAPE FAILURE
53	02	MEDIUM REMOVAL PREVENTED
5A	00	OPERATOR REQUEST OR STATE CHANGE INPUT (UNSPECIFIED)
5A	01	OPERATOR MEDIUM REMOVAL REQUEST
5A	02	OPERATOR SELECTED WRITE PROTECT
5A	03	OPERATOR SELECTED WRITE PERMIT
5B	00	LOG EXCEPTION
5B	01	THRESHOLD CONDITION MET
5B	02	LOG COUNTER AT MAXIMUM
5B	03	LOG LIST CODES EXHAUSTED
70	NN	DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN
71	00	DECOMPRESSION EXCEPTION LONG ALGORITHM ID
80-FF	80-FF	Vendor Unique
ALL CODES NOT SHOWN ARE RESERVED.		

* used on cost sensitive architectural model

Table 6-68 Recommended Sense Key, ASC and ASCQ for REQUEST SENSE Errors

Sense Key	ASC	ASCQ	Description of Error
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET OR ATAPI SOFT RESET OCCURRED

6.8.12. REWIND Command

The REWIND command causes the Device to position to the **beginning-of-partition** in the current partition. Prior to performing the rewind operation, the Device shall ensure that all buffered data and filemarks have been transferred to the medium.

Table 6-69 Rewind Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (01h)							
1	Reserved							Reserved (Immed)
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

Table 6-70 Recommended Sense Key, ASC and ASCQ for REWIND Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
04	15	00	RANDOM POSITIONING ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.13. SPACE Command

The SPACE command is used to position to a relative filemark or to end of data depending on the Count and Code. Both forward and reverse filemark positioning are provided. If the Device does not support the reverse direction, then a SPACE command in the reverse direction shall be terminated with a “Check Condition” and the Sense Key shall be set to ILLEGAL REQUEST.

Table 6-71 Space Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (11h)							
1	Reserved					Code		
2	(MSB)							
3	Count							
4								(LSB)
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

Table 6-72 Space Code

Code	Description
000b	Reserved
001b	Filemarks
010b	Reserved
011b	End of Data
100b-111b	Reserved

When spacing over filemarks, the **Count** field specifies the number of filemarks to be spaced over in the current partition. A positive value N in the **Count** field shall cause forward positioning (toward **end-of-partition**) over N filemarks ending on the **end-of-partition** side of the last filemark. A zero value in the **Count** field shall cause no change of logical position. A negative value -N (two's complement notation) in the **Count** field shall cause reverse positioning (toward **beginning-of-partition**) over N filemarks ending on the **beginning-of-partition** side of the last filemark.

Implementor's Note: *Reverse movement is intended for small changes in position. Some Devices may not support space reverse.*

If **end-of-data** is encountered while spacing over filemarks, “Check Condition” status shall be returned, the Sense Key shall be set to BLANK CHECK, and the Valid bit shall be set to one in the sense data. Additionally, the **EOM** bit shall be set to one if **end-of-data** is encountered at or after **early-warning**. The Information field shall be set to the difference (residue) of the requested count minus the actual number of filemarks spaced over.

If the **end-of-partition** is encountered while spacing forward over filemarks, “Check Condition” status shall be returned, the Sense Key shall be set to MEDIUM ERROR, the EOM bit shall be set to one, and the Valid bit shall be set to one. The Information field shall be set to the difference (residue) of the requested count minus the actual number of filemarks spaced over.

If **beginning-of-partition** is encountered while spacing over filemarks in the reverse direction, "Check Condition" status and the Sense Key to NO SENSE shall be set. The EOM and Valid bits shall be set to one, and the Information field set to the total number of filemarks not spaced over (the requested number of filemarks minus the actual number of filemarks spaced over).

When spacing to **end-of-data**, the count field is ignored. Upon successful completion, the medium shall be positioned such that a subsequent write operation would append to the last logically recorded information.

If **end-of-partition** is encountered while spacing to **end-of-data**, "Check Condition" status shall be returned, the Sense Key shall be set to MEDIUM ERROR, the EOM bit shall be set to one, and the Valid bit shall be set to zero.

Table 6-73 Recommended Sense Key, ASC and ASCQ for SPACE Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
04	15	00	RANDOM POSITIONING ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.14. TEST UNIT READY Command

The TEST UNIT READY command provides a means to check if the Device is ready. This is not a request for a self-test. If the Device would accept an appropriate medium-access command without returning “Check Condition” status, this command shall return a “Good” status. If the Device cannot become operational or is in a state such that Host action is required to make the unit ready, the Device shall return “Check Condition” status with a Sense Key of NOT READY.

Table 6-74 Test Unit Ready Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (00h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

6.8.14.9. Using the TEST UNIT READY Command

The TEST UNIT READY command is useful in that it allows the Host to poll a Device until it is ready without the need to allocate space for returned data. It is especially useful to check cartridge status. ATAPI Devices are expected to respond promptly to indicate the current status of the Device.

Table 6-75 Recommended Sense Key, ASC and ASCQ for TEST UNIT READY Errors

Sense Key	ASC	ASCQ	Description of Error
00	00	00	NO ADDITIONAL SENSE INFORMATION
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
02	04	00	NOT READY - CAUSE NOT REPORTABLE
02	04	01	NOT READY - IN PROGRESS OF BECOMING READY
02	04	02	NOT READY - INITIALIZING COMMAND REQUIRED
02	04	03	NOT READY - MANUAL INTERVENTION REQUIRED
02	04	04	NOT READY - FORMAT IN PROGRESS
02	3A	00	MEDIUM NOT PRESENT
03	-	-	Deferred medium error
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED

6.8.15. WRITE Command

The WRITE command requests that the Device write the data that is transferred from the Host starting at the current position.

Table 6-76 Write Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (0Ah)							
1	Reserved							1 (Fixed)
1	Reserved							
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

The **Transfer Length** specifies the number of fixed-length blocks to be transferred using the current block length reported in the mode parameter block descriptor.

If the **Transfer Length** is zero, no data shall be transferred and the current position shall not be changed. This condition shall not be considered an error. For Devices that maintain a one-to one relationship between physical and logical blocks and have stopped writing due to an unrecoverable write error, the remainder of the buffer can be written less the block in error that is skipped and the data is lost for that block.

If the Device encounters **early-warning** during a WRITE command, the command shall terminate with “Check Condition” status and the EOM and Valid bits shall be set to one in the sense data. If all data that is to be written is successfully transferred to the medium, the Sense Key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. If any data that is to be written cannot be transferred to the medium when **early-warning** is encountered, the Sense Key shall be set to VOLUME OVERFLOW.

In the case of an unrecovered write error, the Valid bit shall be set to one and the Information field shall be set to the difference (residue) of the sum of the pending requested transfer lengths minus the actual number of blocks and filemarks written. The value in the Information field may exceed the **Transfer Length** for the last command. For formats able to restart a write operation, upon termination, the logical position shall be after the unrecovered block.

Implementor's Note: *The Device should ensure that some additional data can be written to the medium (e.g., labels or filemarks) after writing all data stored in the buffers and the first* ***early-warning*** *indication has been returned to the Host.*

Table 6-77 Recommended Sense Key, ASC and ASCQ for WRITE Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
03	0C	00	MEDIUM ERROR:WRITE ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED
07	27	00	WRITE PROTECTED
0D	00	02	VOLUME OVERFLOW:END-OF-PARTITION/MEDIUM DETECTED

6.8.16. WRITE FILEMARK Command

The WRITE FILEMARK command requests that the Device write a filemark to the current position.

Table 6-78 Write Filemark Command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (10h)							
1	Reserved						0 (WSmk)	0 (Immed)
2	0							
3	0							
4	0							Transfer Length
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	Reserved							
11	Reserved							

The Device shall return status as soon as the command descriptor block has been validated. Any buffered data, and/or filemarks shall be written to the medium prior to setting DSC to 1.

A **Transfer Length** of one enables the writing of the filemark. A **Transfer Length** of zero can be also be used to synchronize (flush) any buffered data.

If the Device encounters **early-warning** during WRITE FILEMARK operation, any buffered data or filemarks shall be written to the medium, and a "Check Condition" will exist when DSC is set to 1. If all buffered data, and filemarks are successfully transferred to the medium, the Sense Key will be set to NO SENSE or RECOVERED ERROR, as appropriate.

Implementor's Note: *The Device should ensure that some additional data can be written to the medium (e.g., labels or filemarks) after writing all data stored in the buffers and the first* ***early-warning*** *indication has been returned to the Host.*

In the case of an unrecovered write error, the Valid bit shall be set to one and the Information field shall be set to the difference (residue) of the sum of the pending requested transfer lengths minus the actual number of blocks and filemarks written. The value in the Information field may exceed the **Transfer Length** for the last command.

Table 6-79 Recommended Sense Key, ASC and ASCQ for WRITE FILEMARK Errors

Sense Key	ASC	ASCQ	Description of Error
02	04	00	NOT READY, CAUSE NOT REPORTABLE
02	3A	00	MEDIUM NOT PRESENT
03	0C	00	MEDIUM ERROR:WRITE ERROR
05	20	00	INVALID COMMAND OPERATION CODE
05	24	00	INVALID FIELD IN COMMAND PACKET
06	28	00	NOT READY TO READY TRANSITION
06	29	00	POWER ON, RESET, OR ATAPI SOFT RESET OCCURRED
06	2A	01	MODE PARAMETERS CHANGED
07	27	00	WRITE PROTECTED
0D	00	02	VOLUME OVERFLOW:END-OF-PARTITION/MEDIUM DETECTED

6.9. Glossary for ATAPI Device

Beginning-of-partition. The position at the beginning of the permissible recording region of a partition. If only one partition is defined, this position is typically equivalent to the beginning-of-medium.

beginning-of-medium. The extreme position along the medium in the direction away from the supply reel which can be accessed by the Device.

early-warning. A physical mark or Device computed position near but logically before the **end-of-partition** (independent of physical direction).

end-of-data. End of data in a partition is denoted in format-specific manner. See the **EOD** defined field in the Device configuration page.

end-of-medium. The extreme position along the medium in the direction away from the take-up reel that can be accessed by the Device. This position may be accessed by Devices that support the LOAD UNLOAD command with the **EOT** bit set to one.

end-of-partition. The position at the end of the permissible recording region of a partition.

partition. The entire usable region of recording and reading paths in a volume or in a portion of a volume, defined in a Device-specific manner. If there is more than one partition, they shall be numbered starting with zero (i.e., **beginning-of-partition-zero**).

volume. A recording medium together with its physical carrier.

6.10. Vendor Identification List

Vendor Identification List

ID	Organization
3M	3M Company
ADAPTEC	Adaptec
ADSI	Adaptive Data Systems, Inc. (a Western Digital subsidiary)
AMCODYNE	Amcodyne
ANAMATIC	Anamartic Limited (England)
ANCOT	ANCOT Corp.
APPLE	Apple Computer, Inc.
ARCHIVE	Archive
ASPEN	Aspen Peripherals
AST	AST Research
AT&T	AT&T
BALLARD	Ballard Synergy Corp.
BULL	Bull Peripherals Corp.
CALIPER	Caliper (California Peripheral Corp.)
CAST	Advanced Storage Tech
CDC	Control Data or MPI
CHEROKEE	Cherokee Data Systems
CIE&YED	YE Data, C.Itoh Electric Corp.
CIPHER	Cipher Data Products
Ciprico	Ciprico, Inc.
COMPORT	Comport Corp.
CNGR SFW	Congruent Software, Inc.
COGITO	Cogito
CONNER	Conner Peripherals
CROSFLD	Crosfield Electronics
CYGNET	Cygnet Systems, Inc.
DATACOPY	Datacopy Corp.
DATAPT	Datapoint Corp.
DEC	Digital Equipment
DENON	Denon/Nippon Columbia
DEST	DEST Corp.
DGC	Data General Corp.
DIGIDATA	Digi-Data Corporation
DILOG	Distributed Logic Corp.
DTC	Data Technology Corp.
DPT	Distributed Processing Technology
DXIMAGIN	DX Imaging
EMULEX	Emulex
EPSON	Epson
EXABYTE	Exabyte Corp.
FILENET	FileNet Corp.
FUJITSU	Fujitsu
FUTURED	Future Domain Corp.
GIGATAPE	GIGATAPE GmbH
GIGATRND	GigaTrend Incorporated
Goidelic	Goidelic Precision, Inc.
GOULD	Gould
HITACHI	Hitachi America Ltd or Nissei Sangyo America Ltd
HP	Hewlett Packard
IBM	International Business Machines
IGR	Intergraph Corp.
IMPRIMIS	Imprimis Technology Inc.
IOMEGA	Iomega
ISi	Information Storage inc.

JVC	JVC Information Products Co.
KODAK	Eastman Kodak
KONAN	Konan
KONICA	Konica Japan
LAPINE	Lapine Technology
LASERDRV	LaserDevice Limited
LMS	Laser Magnetic Storage International Company
MATSHITA	Matsushita
MAXTOR	Maxtor Corp.
MELA	Mitsubishi Electronics America
MELCO	Mitsubishi Electric (Japan)
MICROBTX	Microbotics Inc.
MICROP	Micropolis
MICROTEK	Microtek Storage Corp
MINSCRIB	Miniscribe
MOTOROLA	Motorola
NAI	North Atlantic Industries
NatSemi	National Semiconductor Corp.
NCL	NCL America
NCR	NCR Corporation
NEC	NEC
NKK	NKK Corp.
NT	Northern Telecom
OSI	Optical Storage International
OPTIMEM	Cipher/Optimem
OPTOTECH	Optotech
OTL	OTL Engineering
PERTEC	Pertec Peripherals Corporation
PFTI	Performance Technology Inc.
PRAIRIE	PrairieTek
PTI	Peripheral Technology Inc.
PRIAM	Priam
QUALSTAR	Qualstar
QUANTUM	Quantum Corp.
RICOH	Ricoh
RODIME	Rodime
RTI	Reference Technology
SEAGATE	Seagate
SIEMENS	Siemens
SMS	Scientific Micro Systems/OMTI
SONY	Sony Corporation Japan
SPERRY	Sperry (now Unisys Corp.)
STK	Storage Technology Corporation
SUN	Sun Microsystems, Inc.
SYSGEN	Sysgen
T-MITTON	Transmitton England
TANDBERG	Tandberg Data A/S
TANDON	Tandon
TEAC	TEAC Japan
TI-DSG	Texas Instruments
TOSHIBA	Toshiba Japan
UNISYS	Unisys
USDC	US Design Corp.
VERBATIM	Verbatim Corporation
VRC	Vermont Research Corp.
WangDAT	WangDAT
WANGTEK	Wangtek

WDIGTL	Western Digital
XEBEC	Xebec Corporation

❒

---

i

ii